Package ‘psychNET’

July 3, 2019

Type Package

Title Psychometric Networks for Intensive Longitudinal Data

Version 0.0.1

Date 2019-6-21

Description In the past decade, mental processes have been conceptualized as complex networks of interacting psychiatric symptoms. These networks that can be visualized by means of conditional independence graphs. For estimating the underlying directed graph from intensive longitudinal data, vector autoregression (VAR) is the most commonly used tool. This package wraps several methods in the VAR family that can be used to estimate conditional independence graphs (networks) from multivariate time-series data. The package can fit the simple VAR and VARX model Lutkepohl, H. (2005) <doi:10.1007/978-3-540-27752-1> that are currently available from the R package 'vars', and its sparse alternative by Basu S. and Michailidis, G.(2015) <doi:10.1214/15-AOS1315> and sparse VECM implemented in the R package 'sparsevar'. The sparse graphical VAR with covariance estimation by Wild, B., Eichler, M., Friederich, H. C., Hartmann, M., Zipfel, S., & Herzog, W. (2010) <doi:10.1186/1471-2288-10-28> from the R package 'graphicalVAR' and the dynamic factor model by Doz, Gianone & Reichlin (2011) <doi:10.1016/j.jeconom.2011.02.012> from the R package 'dynfactoR' are also available. Sparse estimation of high dimensional VAR, VARMA and VARX models using hierarchical lag structures Nicholson, W. B., Bien, J., Matteson, D. S. (2017) <arXiv:1412.5250v3> implemented in the R package 'bigtime' and mixed VAR for symptom time series with marginal distributions in the exponential family Haslbeck, J., Waldorp, L. J. (2015) <arXiv:1510.06871> from the package 'mgm' can also be used with this package. For the inference of symptom networks from multivariate time series of multiple individuals the 'psychNET' package adopts the multi-level VAR by Epskamp, S., Waldorp, L. J., Mottus, R., & Borsboom, D. (2017) <arXiv:1609.04156v6> implemented in the R package 'mlVAR' and for the high-dimensional setting the sparse time series chain graphical (group graphical VAR) model by Abegaz, F., Wit, E. (2013) <doi:10.1093/biostatistics/kxt005> available from the R package 'sparseTSCGM'.

Depends R (>= 3.5)

Imports vars, igraph, imputeTS, Hmisc, SparseTSCGM, mlVAR, qgraph, graphicalVAR, sparsevar, bigtime, mgm, crayon, longitudinal, networktools, gtools, car, stats, Matrix, methods, MASS, ordinalNet, glmnet, fastDummies

License GPL (>= 2)

URL http://www.math.rug.nl/stat/Main/HomePage

Maintainer Spyros Balafas <s.balafas@rug.nl>

Repository CRAN

NeedsCompilation no

Author Spyros Balafas [aut, cre],
Ernst Wit [aut],
Marco Grzegorczyk [aut],
Sanne Booij [ctb],
Hanneke Wardenaar-Wigman [ctb]

Date/Publication 2019-07-03 15:20:03 UTC

R topics documented:

`psychNET-package` *psychNET : Psychometric network modelling for multivariate time series data.*

Description

This package has been designed in order to provide various psychometric network modelling techniques for multivariate time series data from the behavioral domain, in a simple wrapper function. The aim is to estimate the temporal and contemporaneous interaction structure in a (possibly) sparse fashion and thereby being able to visualize these interactions by means of conditional independence graphs. Depending on the model, the interactions are estimated at the individual level – intra-individual dynamics – or population level – inter-individual dynamics.

Details

In the last decade, time series data became popular in the behavioral sciences. These data, allow us to study the dynamics of complex behavioral systems both at the individual and the population level. Vector autoregression (VAR) is the cornerstone in the statistical modelling of multivariate time series data and various VAR extensions are available that can handle cases where additional complexities are imposed in the analysis.

This package introduces the main `psychNET` function that is used to fit various dynamic models. Models that can be fitted to time-series data from one person using the `psychNET` function are:

- Vector Autoregressive (VAR) models: Traditional VAR model or VAR model with exogenous variables (VARX). See the help file from the `VAR()` function from the R package **vars** for more details.
- Sparse Vector Autoregressive (SVAR) models: SVAR model estimated using penalized multivariate least squares (see `fitVAR` function in the R package **sparsevar**). SVARs estimated using penalized least squares with simultaneous lag estimation (SVARHL) and exogenous variables (SVARHLX) implemented in the functions `sparseVAR` and `sparseVARX` respectively from the R package **bigtime**. SVAR model with simultaneous covariance estimation using penalized likelihood approach known as graphical VAR (GVAR) implemented in the function `graphicalVAR` from the package **graphicalVAR**. A SVAR model for mixed type of time series with marginal distributions in the exponential family (SMVAR), implemented in the function `mvar` from the R package **mgm**.
- Sparse Vector Error Correction Model (SVECM): SVECM estimated using penalized multivariate least squares that is implemented in the function `fitVECM` from the R package **sparsevar**.
- Sparse Vector Autoregressive Moving Average (SVARMA) model: SVARMA model where the parameters are estimated using penalized least squares with simultaneous lag estimation (SVARMAHL) implemented in the function `sparseVARMA` from the R package **bigtime**.
- Sparse Time-Varying Vector Autoregressive (TV-SVAR) model: TV-SVAR model where the parameters are allowed to vary smoothly with time. This model can handle time series with marginal distributions in the exponential family. The model is implemented in the function `tvmvar`, which is available with the R package **mgm**. (This model is NOT yet available in the **psychNET** package. To be expected with the next update.)
- Dynamic factor model (DFM): DFM with three different types of estimates for the latent series. For details you can check the function `dfm` from the R package **dynfactoR** available at github.

For network inference at the population level from nested time-series data, two models in the broad class of VAR models can be fitted via the `psychNET` function. These are:

- Multilevel Vector Autoregressive (MLVAR) model: This model estimates group as well as individual level effects and it is implemented in the R function **mlVAR** from the R package **mlVAR**.
- Sparse Vector Autoregressive (SVAR) model: This model estimates sparse group level networks from nested time series with simultaneous covariance estimation. We call this model group graphical VAR (GGVAR) and it has been primarily implemented in the function `sparse.tscgm` available with R package **SparseTSCGM**.

Author(s)

Spyros E. Balafas (author, creator), Sanne Booij, Marco A. Grzegorczyk, Hanneke Wardenaar-Wigman, Ernst C. Wit

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

Lutkepohl, H. (2006). *New Introduction to Multiple Time Series Analysis.* Springer, New York.

Basu, S., Michailidis, G. (2015). *Regularized estimation in sparse high-dimensional time series models.* Ann. Statist. 43, no. 4, 1535-1567.

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

Haslbeck, J., Waldorp, L. J. (2016). *mgm: Structure Estimation for time-varying Mixed Graphical Models in high-dimensional Data.*

Nicholson, W. B., Bien, J., Matteson, D. S. (2017). *High Dimensional Forecasting via Interpretable Vector Autoregression.*.

Wilms, I., Basu, S., Bien, J., Matteson D. S. (2017). *Sparse Identification and Estimation of High-Dimensional Vector AutoRegressive Moving Averages.*

Epskamp, S., Waldorp, L. J., Mottus, R., Borsboom, D. (2016). *The Gaussian Graphical Model in Cross-sectional and Time-series Data.*

Examples

```
# Load the R package psychNET
library(psychNET)
```

`plot.pnt` *Plot method for S3 class "pnt" objects.*

Description

Generic that plots an object of class "pnt".

Usage

```
## S3 method for class 'pnt'
plot(x, type, person, community, ...)
```

Arguments

`x`	: An object resulted from the `psychNET` function.
`type`	: String argument, which controls the type of plot that will be returned. The available options are: "temporal" (for temporal network), "contemporaneous" (for contemporaneous network), "between" (for between subjects network), and "both" (both temporal and contemporaneous networks)
`person`	: This can be a single number or vector of numbers that denotes the person index for which plots will be returned. This argument is used only when the model fitted is a Multi-level VAR (i.e. the argument `model` in the `psychNET` function equals to "MLVAR".)
`community`	: Logical argument. When `TRUE`, the function fits a spinglass community detection algorithm with negative weights where only the present edges are taken into account. !!WARNING!! this can make the `plot.pnt` function to be very slow.
`...`	: Other arguments to be passed on to the `plot.igraph` function. Use this with care since some arguments in are already specified internally.

Details

For This function is used to visualize networks estimated via the `psychNET` function. See additional details of the function `psychNET`.

Value

The value returned by the plot is a list where its elements are two lists named temporal and contemporaneous respectively. These lists contain objects of class `igraph` and `qgraph` that can be used by the user to create tailor made plots.

Author(s)

Spyros E. Balafas (author, creator), Sanne Booij, Marco A. Grzegorczyk, Hanneke Wardenaar-Wigman, Ernst C. Wit

Maintainer: Spyros E. Balafas (`<s.balafas@rug.nl>`)

References

Lutkepohl, H. (2006). *New Introduction to Multiple Time Series Analysis.* Springer, New York.

Basu, S., Michailidis, G. (2015). *Regularized estimation in sparse high-dimensional time series models.* Ann. Statist. 43, no. 4, 1535-1567.

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

Haslbeck, J., Waldorp, L. J. (2016). *mgm: Structure Estimation for time-varying Mixed Graphical Models in high-dimensional Data.*

Nicholson, W. B., Bien, J., Matteson, D. S. (2017). *High Dimensional Forecasting via Interpretable Vector Autoregression..*

Wilms, I., Basu, S., Bien, J., Matteson D. S. (2017). *Sparse Identification and Estimation of High-Dimensional Vector AutoRegressive Moving Averages.*

Epskamp, S., Waldorp, L. J., Mottus, R., Borsboom, D. (2016). *The Gaussian Graphical Model in Cross-sectional and Time-series Data.*

`print.pnt` *Print objects of class "pnt".*

Description

Generic `print` function that is used to print information to the console for the function call, the time elapsed, the data used, the fitted model and the model estimates.

Usage

```
## S3 method for class 'pnt'
print(x, ...)
```

Arguments

`x`	: is an object resulted from the `psychNET` function
`...`	: Not used

Author(s)

Spyros E. Balafas (author, creator), Sanne Booij, Marco A. Grzegorczyk, Hanneke Wardenaar-Wigman, Ernst C. Wit

Maintainer: Spyros E. Balafas (`<s.balafas@rug.nl>`)

References

Lutkepohl, H. (2006). *New Introduction to Multiple Time Series Analysis.* Springer, New York.

Basu, S., Michailidis, G. (2015). *Regularized estimation in sparse high-dimensional time series models.* Ann. Statist. 43, no. 4, 1535-1567.

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

Haslbeck, J., Waldorp, L. J. (2016). *mgm: Structure Estimation for time-varying Mixed Graphical Models in high-dimensional Data.*

Nicholson, W. B., Bien, J., Matteson, D. S. (2017). *High Dimensional Forecasting via Interpretable Vector Autoregression.*.

Wilms, I., Basu, S., Bien, J., Matteson D. S. (2017). *Sparse Identification and Estimation of High-Dimensional Vector AutoRegressive Moving Averages.*

Epskamp, S., Waldorp, L. J., Mottus, R., Borsboom, D. (2016). *The Gaussian Graphical Model in Cross-sectional and Time-series Data.*

See Also

`psychNET`

`psychNET` *Psychometric networks estimated by multivariate time series methods.*

Description

Wrapper function of multivariate time-series models, which can be used to obtain symptom networks (i.e., networks of symptom-symptom interactions).

Usage

```
psychNET(data, model, lag, criterion, nFact, penalty, lambda1, lambda2,
covariates, impute, transform, ...)
```

Arguments

`data` : Usually a "matrix", "data.frame", or "longitudinal" object of dimension *(n*T) x p* where *n* is the number of individuals. For time-series data from multiple persons, an additional numeric column explicitly named "ID" indicating from which person the measurements are coming from should be included in the data. Additionally, with experience sampling data that are nested within days, additional columns named "DAY" and "BEEP" can be included in data from which we calculate the consecutiveness of the measurements. Alternatively, a column named "TIME" denoting the measuremnt consecutiveness can be provided in data. See details.

`model` : This argument controls the model to be fitted to the data. The available options are: "VAR", "SVAR", "SMVAR", "SVECM", "GVAR", "SVARHL", "SVARHLX", "SVARHLMA", "DFM", "MLVAR", "GGVAR".

`lag` : Specifies the lag order of the process. When model is "DFM" lag corresponds to the order of the factor process. When model is "SVARHL", "SVARHLX", or "SVARHLMA" lag can be NULL and the model chooses the optimal lag via regularized hierarchical lag structures.

`criterion` : The information criterion to be used in order to tune the penalty parameters of regularized (sparse) VAR models. Available options are: "CV", "AIC", "BIC", "EBIC", "GIC", and "MBIC". This argument depends on the model argument (see details).

`nFact` : This argument is used only when model="DFM" and controls the number of the static factors in the dynamic factor model.

`penalty` : This argument controls the type of regularization to be used and depends on the model argument (see details). Available options are "LASSO", "ENET", "SCAD", "MCP", and "HLag".

`lambda1` : A numeric vector with length greater than one that specifies the values of the penalty parameter for regularized VARs. Typically, this corresponds to the penalty parameter for the lagged effects. When model is "GVAR" or "GGVAR" then lambda1 is used as the penalty value for the precision matrix.

`lambda2` : A numeric vector with length greater than one. This argument is used only when model is "GVAR" or "GGVAR" and controls the value of the penalty parameter for the lagged effects.

`covariates` : This argument is used only when model is "VAR" or "SVARHLX" in order to specify covariates for VARX or SVARX models.

`impute` : String that is used to specify the imputation method when missing values exist in the data or when missing values are inserted to transform non-equidistant measurements into equidistant. It can be a single string or a vector of strings with length equal to the number of symptoms in the data. In the latter case, each symptom can be imputed using an explicit method. Available options are: "Kalman.arima", "Kalman.struct", "Interpol.linear", "Interpol.spline", "Interpol.stine", "LOCF", "NOCB", "MA.simple", "MA.linear", "MA.exponent", "mean", "mode", "median", "random", "Season.int.linear", "Season.int.spline", "Season.int.stine", "Season.LOCF", "Season.NOCB","Season.mean", "Season.median", "Season.mode", "Season.MA", "Season.kalman", "Season.random".

`transform` : String that is used to specify the transormation function. It can be a single string or a vector of strings with length equal to the number of symptoms in the data. In the latter case, each symptom can be transformed using an explicit function. Available options are: `"log"`, `"log10"`, `"Copula_discr"`, `"Copula_skew"`, `"Zero.mean"`, `"Standardize"`, `"Power"`, `"Logit"`, `"Square.root"`, `"Power2"`, `"Power3"`, `"Cube.root"`.

`...` Argument that depends on the value of the `model` argument. It is used to pass additional arguments to the model fitting function.

Details

The data

The data to be used in the `psychNET` function must strictly contain symptom expression data where the measurements are ordered by time. The only additional variables that are allowed in the data except symptom expression are a variable explicitly coded as "ID" when multiple persons need to be analyzed together, a variable coded as "DAY" when multiple measurements per day are taken and a variable called "BEEP" to indicate the measurement intensity within a given day. The variables "DAY" and "BEEP" are used internally to construct a time variable that denotes the consecutiveness of the measurements. The time variable can also be provided in the data explicitly named as "TIME". The only models in the package that can handle non-equidistant observations naturally are the `"GVAR"`, `"SMVAR"`, and `"MLVAR"`. In any other model, non-equidistant measurements first are transformed to equidistant by inserting missing values, which are then imputed using the imputation method specified in the `impute` argument.

The models

Here we provide some more details with respect to the models implemented in the package.

VAR model **(VAR)**

The traditional VAR model is fitted with the `psychNET` function by setting `model="VAR"`. This model is originally implememnted in the function `VAR` that comes with the R package **vars**. In this implementation the unknown model parameters are estimated by ordinary least squares (OLS) per equation. The arguments `lag` and `covariates` of the `psychNET` function correspond to the parameter `p` and `exogen` respectively of the original `VAR` function. Other parameters that are used in the original function can be passed through the `...` structure (technically known as ellipsis). For details on arguments that can be passed to the three dots when fitting a VAR the user can type `?vars::VAR` into the console in order to access its help file.

Sparse VAR model **(SVAR)**

A sparse VAR model is fitted with the `psychNET` function by setting `model="SVAR"`. This model is implememnted in the function `fitVAR` from the R package **sparsevar** and the unknown model parameters are estimated via penalized multivariate least squares. Three different types of penalization are available that can be passed to the function through the `penalty` argument. If you set `penalty="ENET"` an elasticnet penalty is applied to the parameters with other option being `"SCAD"` for SCAD type of penalization and `"MCP"` for MC+ penalty. The only option available for tuning the penalty parameter is k-fold cross validation. The parameter `lag` of the `psychNET` function corresponds to the parameter `p` of the `fitVAR` function, the parameter `penalty` is used in the same way as in the `fitVAR` function while `criterion` substitutes the argument `method` of the `fitVAR` function. Additional arguments can be passed to the `fitVAR` function by using the `...` structure of the psychNET function. For details on the original function and additional parameters that can be used you can look at ?sparsevar::fitVAR.

***Sparse VAR model* with hierarchical lag structures and additional covariates (SVARHL and SVARHLX)**

A sparse VAR that offers the possibility of simultaneous lag estimation is fitted by setting model="SVARHL". Additional covariates can be used by setting model="SVARHLX" and providing the covariates data to the argument covariates of the psychNET function. These two models are available with the R package **bigtime** in the functions sparseVAR and sparseVARX respectively. These models are estimated by penalized least squares optimized by a proximal gradient algorithm. These models offer two types of penalization and the tuning of the penalty parameters is done by k-fold cross validation. One option is to penalize the VAR coefficients by hierarchical group LASSO regularization and the other option is the standard LASSO also known as L1 regularization. For hierarchical group LASSO the user must set penalty="HLag" and for standard LASSO penalty="LASSO". In the psychNET function, penalty argument corresponds to the VARpen and VARXpen arguments of the functions sparseVAR and sparseVARX respectively, while lag is associated with the parameter p of these two functions. Typically, the function internally calculates a grid for the regularization parameter corresponding to sparse penalty. The user can provide her/his own grid of regularization parameters with the argument lambda1. The latter acts like the arguments VARlseq and VARXlPhiseq of the original functions sparseVAR and sparseVARX respectively. Other arguments of these functions can be passed via the ... structure of the psychNET function. If you want the function to estimate the lag of the process simultaneously you must set lag=NULL. For details on the original functions and additional parameters that can be used you can look at ?bigtime::sparseVAR and ?bigtime::sparseVARX.

***Sparse VAR model* with simultaneous covariance estimation (GVAR)**

A sparse VAR with simultaneous covariance estimation (also known as graphical VAR) is fitted by setting model="GVAR". This model is available with the R package **graphicalVAR** in the function graphicalVAR. The model parameters are estimated by otimizing a penalized maximum likelihood. The only penalization option when model="GVAR" is LASSO, which means that penalty argument must be equal to "LASSO". The tuning of the penalty parameters is done by the extended Bayesian information criterion (EBIC) by setting criterion="EBIC". Model selection via standard BIC is possible by using an additional argument gamma=0 in the psychNET function. This model uses two distinct penalties, one for the autoregressive parameters and one for the covariance parameters. Typically, the function calculates its own grid of penalty parameters for the regularization, however, the user can provide her/his own grid by using the parameters lambda1 (for covariance penalization) and lambda2 (for autoregressive penalization) respectively in the function psychNET. Other arguments of the original graphicalVAR function can be passed via the ... structure of the psychNET function.

!!WARNING!! Do not use the ... to provide the following arguments of the graphicalVAR function: vars, scale, idvar, beepvar, dayvar, centerWithin, deleteMissings since they are already specified internally. If you want to use idvar, beepvar, dayvar arguments of the original function graphicalVAR, these variables must be provided as columns in the data object explicitly named as "ID", "BEEP", and "DAY" respectively. For scale and centerWithin arguments you can use the argument transform of the psychNET function. For details on the original function and additional parameters that can be used you can look at ?graphicalVAR::graphicalVAR.

***Sparse VAR model* for mixed type of time series (SMVAR)**

A sparse VAR with simultaneous covariance estimation (also known as graphical VAR) is fitted by setting model="SMVAR". This model is available with the R package **mgm** in the function mvar. The model parameters are estimated by otimizing a penalized least squares function per equation. The only penalization option when model="SMVAR" is elasticnet, which means that penalty argument must be equal to "ENET". The tuning of the penalty parameters is done by the extended Bayesian information criterion (EBIC) by setting criterion="EBIC" or k-fold cross validation by setting criterion="CV". Typically, the function internally calculates a grid for the regularization parameter. The user can provide her/his own grid of regularization parameters with the argument lambda1. The latter acts analogous to the argument lambdaSeq of the original mvar function. Model selection via standard BIC is possible by using an additional argument gamma=0 in the psychNET function. Other arguments of the original mvar function can be passed via the ... structure of the psychNET function.

!!WARNING 1!! When model="SMVAR" the data must be given in a "data.frame" format with the following properties:

1. Gaussian variables: must be of class "numeric"
2. Poisson variables: must be of class "integer"
3. Categorical variables: must be of class "factor"
4. Ordinal variables: must be of class "ordered factor" (SMVAR for ordinal type of variables to be expected in a following version of the package **psychNET**).

!!WARNING 2!! Do not use the ... to provide the following arguments of the mvar function: type, level, scale, beepvar, dayvar since they are already specified internally. If you want to use beepvar and dayvar arguments of the original function mvar, these variables must be provided as columns in the data object explicitly named as "BEEP", and "DAY" respectively. For scale you can use the argument transform of the psychNET function. For details on the original function and additional parameters that can be used you can look at ?mgm::mvar.

***Sparse VEC model* (SVECM)**

A sparse VEC (vector error correction) model is fitted with the psychNET function by setting model="SVECM". This model is implememnted in the function fitVECM from the R package **sparsevar** and the unknown model parameters are estimated via penalized multivariate least squares. Three different types of penalization are available that can be passed to the function through the penalty argument. If you set penalty="ENET" an elasticnet penalty is applied to the parameters with other option being "SCAD" for SCAD type of penalization and "MCP" for MC+ penalty. The only option available for tuning the penalty parameter is k-fold cross validation. The parameter lag of the psychNET function corresponds to the parameter p of the fitVECM function, the parameter penalty is used in the same way as in the fitVECM function while criterion substitutes the argument method of the fitVAR function. Additional arguments can be passed to the fitVAR function by using the ... structure of the psychNET function. For details on the original function and additional parameters that can be used you can look at ?sparsevar::fitVECM. The argument logScale of the orgiginal function is set to FALSE, but you can use the argument transform of the psychNET function instead.

***Sparse VARMA model* with hierarchical lag structures (SVARHLMA)**

A sparse VARMA (vector autoregressive moving average) model that offers the possibility of simultaneous lag estimation is fitted by setting model="SVARHLMA". This model is available with the R package **bigtime** in the function sparseVARMA. The model parameters are estimated by penalized least squares optimized by a proximal gradient algorithm. The model offers two types of penalization and the tuning of the penalty parameters is done by k-fold cross validation. One option is to penalize the VAR coefficients by hierarchical group LASSO regularization and the other option is the standard LASSO. For hierarchical group LASSO the user must set `penalty="HLag"` and for standard LASSO `penalty="LASSO"`. In the `psychNET` function, the `penalty` argument corresponds to the `VARpen` argument of the function `sparseVARMA`, while `lag` is associated with the parameter `p` of this function. Typically, the function internally calculates a grid for the regularization parameter corresponding to sparse penalty. The user can provide her/his own grid of regularization parameters with the argument `lambda1`. The latter acts like the arguments `VARlseq` of the original function `sparseVARMA`. Other arguments of this function can be passed via the `...` structure of the `psychNET` function. If you want the function to estimate the lag of the process simultaneously you must set `lag=NULL`. For details on the original function and additional parameters that can be used you can look at `?bigtime::sparseVARMA`.

***Dynamic factor model* (DFM)**

A DFM is fitted by setting `model="DFM"` in the `psychNET` function. This model is available with the R package **dynfactoR** (available at github) in the function `dfm`. The model parameters are estimated by an EM algorithm. The argument `nFact` of the `psychNET` function is used to specify the number of static factors and it is analogous to the argument `r` of the original `dfm` function. The argument `lag` of the `psychNET` function is used to specify the lag order of the factor process analogously to the argument `p` of the `dfm` function. The DFM that we implement here is restricted to have an identity system state covariance matrix and zeros at the upper diagonal elements of the factor loading matrix for identifiability purposes. The user of the **psychNET** package can specify her/his own threshold for algorithm convergence by using an additional argument `threshold`.

***Multi-level VAR model* for group level dynamics (MLVAR)**

An MLVAR can be fitted by setting `model="MLVAR"`. This model is available with the R package **mlVAR** in the function `mlVAR`.

!!WARNING!! Do not use the `...` to provide the following arguments of the `mlVAR` function: `vars`, `scale`, `idvar`, `beepvar`, `dayvar`, `estimator`, and `scaleWithin` since they are already specified internally. If you want to use `idvar`, `beepvar`, `dayvar` arguments of the original function `mlVAR`, these variables must be provided as columns in the `data` object explicitly named as "ID", "BEEP", and "DAY" respectively. For the `scaleWithin` argument you can use the argument `transform` of the `psychNET` function. For details on the original function and additional parameters that can be used you can look at `?mlVAR::mlVAR`. The only estimation possible is "lmer".

***Sparse VAR model* with simultaneous covariance estimation for group level dynamics (GGVAR)**

A sparse VAR with simultaneous covariance estimation for multiple individuals (also known as graphical VAR) is fitted by setting `model="GGVAR"`. This model is available with the R package **SparseTSCGM** in the function `sparse.tscgm`. The model parameters are estimated by optimizing a penalized maximum likelihood. Penalization options when `model="GGVAR"` is LASSO and SCAD, which means that `penalty` argument must be equal to `"LASSO"` for LASSO penalization and equal to `"SCAD"` for SCAD penalization. The tuning of the penalty parameters is done by AIC, BIC, EBIC, GIC (generalized information criterion), and MBIC (modified Bayesian information criterion) by using the `criterion` argument which is the same as the `optimality` argument in the `sparse.tscgm` function. This model uses two distinct penalties, one for the autoregressive parameters and one for the covariance parameters. Typically, the function calculates its own grid of penalty parameters for the regularization, however, the user can provide her/his own grid by using the parameters `lambda1` (for covariance penalization) and `lambda2` (for autoregressive penalization) respectively in the function `psychNET`. Other arguments of the original `graphicalVAR` function can be passed via the `...` structure of the `psychNET` function. Be aware that only VAR models with up to 2 lags are possible.

Additional details The psychNET function will estimate (based on the lag of the process) one or more temporal conditional independence graphs unless the model equals to "DFM". For model="DFM" and lag=1 the function will estimate the equivalent VAR(1), whereas for lag greater than one, the temporal network of the factor interactions will be estimated. Contemporaneous conditional independence graphs are available only when the model argument equals to: "GVAR", "GGVAR", "MLVAR", or "DFM" with one lag.

Value

The function psychNET returns a list object of class pnt.

Author(s)

Spyros E. Balafas (author, creator), Sanne Booij, Marco A. Grzegorczyk, Hanneke Wardenaar-Wigman, Ernst C. Wit

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

Lutkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis.* Springer, New York.

Basu, S., Michailidis, G. (2015). *Regularized estimation in sparse high-dimensional time series models.* Ann. Statist. 43, no. 4, 1535-1567.

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

Haslbeck, J., Waldorp, L. J. (2015). *mgm: Structure Estimation for time-varying Mixed Graphical Models in high-dimensional Data.*

Nicholson, W. B., Bien, J., Matteson, D. S. (2017). *High Dimensional Forecasting via Interpretable Vector Autoregression.*.

Wilms, I., Basu, S., Bien, J., Matteson D. S. (2017). *Sparse Identification and Estimation of High-Dimensional Vector AutoRegressive Moving Averages.*

Epskamp, S., Waldorp, L. J., Mottus, R., Borsboom, D. (2016). *The Gaussian Graphical Model in Cross-sectional and Time-series Data.*

Examples

```
## load the psychNET library
library(psychNET)

## load the Canada dataset from the 'vars' package
data("Canada", package = "vars")
Canada_data_frame <- data.frame(Canada)

## fit a VAR model
VAR_model <- psychNET(Canada_data_frame, model = "VAR", lag = 1, type = "const")
# print the result
VAR_model
# summarize the resulting network
summary(VAR_model)
```

```
# summarize the VAR model using the original summary method
vars:::summary.varest(VAR_model$fit)
# plot the VAR model results using the original plot method
vars:::plot.varest(VAR_model$fit)
# plot the resulting network
plot(VAR_model)

## fit a sparse VAR model
sparse_VAR_model <- psychNET(Canada_data_frame, model = "SVAR", lag = 1)
# print the result
sparse_VAR_model
# summarize the resulting network
summary(sparse_VAR_model)
# plot the sparse VAR model results using the original plot method
sparsevar::plotVAR(sparse_VAR_model$fit)
# plot the resulting network
plot(sparse_VAR_model)

## fit a sparse VAR model as the one in the 'bigtime' package
sparse_lassVAR_model <- psychNET(Canada_data_frame,
model = "SVARHL",penalty = "LASSO", lag = 1, VARgran=c(500,1000))
# print the result
sparse_lassVAR_model
# summarize the resulting network
summary(sparse_lassVAR_model)
# plot the resulting network
plot(sparse_lassVAR_model)


## Load the psychNET package
library(psychNET)
#################################### N=1 models #####################################
#####################################################################################


###################################################################
######## REPRODUCE EXAMPLE OF A VAR FROM THE 'vars' PACKAGE ########
###################################################################

## load the 'vars' package
library(vars)

## Load the Canada dataset from the vars package
data(Canada)

## check the structure of the data
str(Canada)

## The data is in time series format. It needs to be transformed into
## a matrix, data.frame or longitudinal object for the psychNET package
Canada_data_frame <- data.frame(Canada)

## fitting a VAR using the vars package
```

```
varmod <- vars::VAR(Canada, p = 2, type = "none")

## fitting the same VAR using the psychNET package
psychvar <- psychNET(Canada_data_frame, model = "VAR", lag = 2, type = "none")

## Check if the results are the same
all.equal(Acoef(varmod), psychvar$results$A, check.attributes = FALSE)

################################################################################
####### Fit A DYNAMIC FACTOR TO THE CANADA DATA FROM THE 'vars' PACKAGE  #######
################################################################################

## install and load the 'devtools' package and the
## by using install.packages("devtools") and then
## library(devtools)
## install the 'dynfactoR' package available on github
## by using devtools::install_github("rbagd/dynfactoR")
## library(dynfactoR)

## Load the Canada dataset from the vars package
data(Canada, package = "vars")

## check the structure of the data
str(Canada)

## The data is in time series format. It needs to be transformed into
## a matrix, data.frame or longitudinal object for the psychNET package
Canada_data_frame <- data.frame(Canada)

## fitting a DFM using the dynfactoR package
dfmmod <- psychNET:::dfm(Canada_data_frame, r=2, p = 1,
                         rQ= "identity", rC= "upper",max_iter = 100000)

## fitting the same DFM using the psychNET package
psychdfm <- psychNET(Canada_data_frame, model = "DFM", nFact = 2,  lag = 1)

## Check if the results are the same
all.equal(dfmmod$A, psychdfm$results$A_fact[[1]], check.attributes = FALSE)
all.equal(dfmmod$C, psychdfm$results$B_fac_symptoms, check.attributes = FALSE)
all.equal(dfmmod$Q, psychdfm$results$System_Covariance, check.attributes = FALSE)
all.equal(dfmmod$R, psychdfm$results$Obs_Covariance, check.attributes = FALSE)


#####################################################################################
########### FIT A SPARSE VAR TO THE CANADA DATA FROM THE 'vars' PACKAGE  ############
#####################################################################################

## load the 'sparsevar' package
library(sparsevar)

## Load the Canada dataset from the vars package
data(Canada, package = "vars")
```

```
## The data is in time series format. It needs to be transformed into
## a matrix, data.frame or longitudinal object for the psychNET package
Canada_data_frame <- data.frame(Canada)

## fitting a SVAR using the sparsevar package
set.seed(1)
svarmod <- fitVAR(Canada, p = 1, penalty="SCAD", method="cv", logScale=FALSE)

## fitting the same SVAR using the psychNET package
set.seed(1)
psychsvar <- psychNET(Canada_data_frame, model = "SVAR",penalty = "SCAD", lag = 1, criterion="CV")

## Check if the results are the same
all.equal(svarmod$A, psychsvar$results$A, check.attributes = FALSE)


#########################################################################################
####### EXAMPLE OF A SPARSE VAR WITH HIERARCHICAL LAGS FROM THE 'bigtime' PACKAGE ######
#########################################################################################

## load the 'bigtime' package
library(bigtime)
## Load the Y dataset from the bigtime package
data(Y, package = "bigtime")

## fitting a SVAR with hierarchical lags
## using the bigtime package
svarhlmod <- sparseVAR(Y, VARpen ="HLag")

## fitting the same model using the psychNET package
psychsvarhl <- psychNET(Y, model = "SVARHL", penalty = "HLag", criterion="CV")

## Check if the results are the same
all.equal(svarhlmod$Phihat, psychsvarhl$fit$Phihat, check.attributes = FALSE)



#######################################################################################
########### REPRODUCE EXAMPLE OF A SPARSE MIXED VAR FROM THE 'mgm' PACKAGE ###########
#######################################################################################

## load the 'mgm' package
library(mgm)

# 1) Define mVAR model as in the mgm manual
p <- 6 # Six variables
type <- c("c", "c", "c", "c", "g", "g") # 4 categorical, 2 gaussians
level <- c(2, 2, 4, 4, 1, 1) # 2 categoricals with m=2, 2 categoricals with m=4, two continuous
max_level <- max(level)
lags <- 1:3 # include lagged effects of order 1-3
n_lags <- length(lags)

# Specify thresholds
```

```
thresholds <- list()
thresholds[[1]] <- rep(0, level[1])
thresholds[[2]] <- rep(0, level[2])
thresholds[[3]] <- rep(0, level[3])
thresholds[[4]] <- rep(0, level[4])
thresholds[[5]] <- rep(0, level[5])
thresholds[[6]] <- rep(0, level[6])

# Specify standard deviations for the Gaussians
sds <- rep(NULL, p)
sds[5:6] <- 1

# Create coefficient array
coefarray <- array(0, dim=c(p, p, max_level, max_level, n_lags))

# a.1) interaction between continuous 5<-6, lag=3
coefarray[5, 6, 1, 1, 2] <- .4
# a.2) interaction between 1<-3, lag=1
m1 <- matrix(0, nrow=level[2], ncol=level[4])
m1[1,1:2] <- 1
m1[2,3:4] <- 1
coefarray[1, 3, 1:level[2], 1:level[4], 1] <- m1
# a.3) interaction between 1<-5, lag=9
coefarray[1, 5, 1:level[1], 1:level[5], 3] <- c(0, 1)


# 2) Sample
set.seed(1)
dlist <- mvarsampler(coefarray = coefarray,
                     lags = lags,
                     thresholds = thresholds,
                     sds = sds,
                     type = type,
                     level = level,
                     N = 200,
                     pbar = TRUE)

# 3) Transform data into a data.frame for psychoNET suitability
# each categorical variable is coded as factor, each poisson as integer, gaussian as numeric
d1 <- as.data.frame(dlist$data)
d1$V1 <- as.factor(d1$V1)
d1$V2 <- as.factor(d1$V2)
d1$V3 <- as.factor(d1$V3)
d1$V4 <- as.factor(d1$V4)
dat <- d1

## fitting the SMVAR model using the mgm package
smvarmod <- mvar(data = dlist$data,
                 type = type,
                 level = level,
                 lambdaSel = "EBIC",
                 lags = 1:3,
                 scale = FALSE,
```

```
                 signInfo = FALSE,
                 overparameterize = FALSE)

## fitting the same model using the psychNET package
psychsmvar <- psychNET(dat, model = "SMVAR",
                       lag = 3,
                       criterion = "EBIC",
                       signInfo = FALSE,
                       overparameterize = FALSE)


all.equal(smvarmod$wadj,psychsmvar$fit$wadj)


################################### N>1 models ########################################
#######################################################################################

######################################################################################
########### REPRODUCE EXAMPLE OF A MULTILEVEL VAR FROM THE 'mlVAR' PACKAGE ###########
######################################################################################

## load the 'mlVAR' package
library(mlVAR)

## Simulate data:
Model <- mlVARsim(nPerson = 50, nNode = 3, nTime = 50, lag=1)

# Estimate an MLVAR with correlated random effects using the mlVAR package
mlvarmod <- mlVAR(Model$Data, vars = Model$vars,
          idvar = Model$idvar, lags = 1, temporal = "correlated")

# Estimate the same MLVAR using the psychNET package
psychmlvar <- psychNET(Model$Data, model="MLVAR",  lag = 1, temporal = "correlated")

# Check if the results are equal
all.equal(mlvarmod$results, psychmlvar$fit$results)

#######################################################################################
########### REPRODUCE EXAMPLE OF A GGVAR VAR FROM THE 'SparseTSCGM' PACKAGE ###########
#######################################################################################

## load the 'SparseTSCGM' package
library(SparseTSCGM)

## Simulate data:
seed = 321
datas <- sim.data(model="ar1", time=10,n.obs=10, n.var=5,seed=seed,
                  prob0=0.35, network="random")
data.fit <-  datas$data1

# Estimate a group graphical VAR (also known as time series chain graphical model)
# using the SparseTSCGM package
ggvarmod <- sparse.tscgm(data=data.fit,
```

```
                         lam1=NULL, lam2=NULL, nlambda=NULL,
                         model="ar1", penalty="scad",optimality="bic",
                         control=list(maxit.out = 10, maxit.in = 100))

# Estimate the same model using the psychNET package
psychggvar <- psychNET(data.fit, model="GGVAR", lag=1, penalty="SCAD", criterion="BIC",
                       control=list(maxit.out = 10, maxit.in = 100))

# Check if the results are equal
all.equal(ggvarmod$theta, psychggvar$fit$theta, check.attributes = FALSE)
all.equal(ggvarmod$gamma, psychggvar$fit$gamma, check.attributes = FALSE)
```

`summary.pnt` *Summary method for S3 class "pnt" objects.*

Description

Generic function that is used in order to summarize information from "pnt" class objects.

Usage

```
## S3 method for class 'pnt'
summary(object, ...)
```

Arguments

`object` : An object obtained by the `psychNET` function.

`...` : Not used in this version of the package.

Details

This a generic function that summarize the information of the networks obtained after a time series model has been fitted to data. Since the main function `psychNET` is a wrapper of several models the summary methods of each method can also be used (if available) by typing `summary(object$fit)` where `object` is an object obtained by the `psychNET` function.

Value

The function `summary.pnt` returns a list with the following components:

`contemporaneous`
: When estimates of the covariance matrix (contemporaneous network) are available then contemporaneous is a list with two components named local and global that describe the contemporaneous network locally and globally. For global description of the network, transitivity, reciprocity, distance, density, and diameter are returned in a one column matrix. At the local (nodes) level the function calculates node transitivity, the degree centrality, step-1 and -2 node expected influence, betweeness centrality and closeness centrality.

`temporal` A list with two components named local and global that describe the temporal network locally and globally. At the global graph level the same descriptives as in the contemporaneous network are returned for each lag of the VAR model. At the local level, node transitivity, in and out degree centrality, step -1 and -2 expected influence centralities, betweeness, out and in closeness centralities are returned to the user.

Author(s)

Spyros E. Balafas (author, creator), Sanne Booij, Marco A. Grzegorczyk, Hanneke Wardenaar-Wigman, Ernst C. Wit

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

Lutkepohl, H. (2006). *New Introduction to Multiple Time Series Analysis.* Springer, New York.

Basu, S., Michailidis, G. (2015). *Regularized estimation in sparse high-dimensional time series models.* Ann. Statist. 43, no. 4, 1535-1567.

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

Haslbeck, J., Waldorp, L. J. (2016). *mgm: Structure Estimation for time-varying Mixed Graphical Models in high-dimensional Data.*

Nicholson, W. B., Bien, J., Matteson, D. S. (2017). *High Dimensional Forecasting via Interpretable Vector Autoregression.*.

Wilms, I., Basu, S., Bien, J., Matteson D. S. (2017). *Sparse Identification and Estimation of High-Dimensional Vector AutoRegressive Moving Averages.*

Epskamp, S., Waldorp, L. J., Mottus, R., Borsboom, D. (2016). *The Gaussian Graphical Model in Cross-sectional and Time-series Data.*

Index