Package ‘PNAR’

November 2, 2022

Type Package

Title Poisson Network Autoregressive Models

Version 1.3

Date 2022-11-02

Author Michail Tsagris [aut, cre],
Mirko Armillotta [aut, cph],
Konstantinos Fokianos [aut]

Maintainer Michail Tsagris <`mtsagris@uoc.gr`>

Description Quasi likelihood-based methods for estimating Poisson Network Autoregression with p lags, PNAR, following generalized linear models are provided. PNAR models with the identity and with the logarithmic link function are allowed. The inclusion of exogenous covariates is also possible. Moreover, it provides tools for testing the linearity of linear PNAR model versus several nonlinear alternatives. Finally, it allows generating multivariate count distributions, from linear and nonlinear PNAR models, where the dependence between Poisson random variables is generated by suitable copulas. References include: Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. <arXiv:2104.06296>. Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. <arXiv:2202.03852>.

Depends R (>= 4.0)

Imports doParallel, foreach, igraph, nloptr, parallel, Rfast, Rfast2,
stats

License GPL (>= 2)

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2022-11-02 08:52:53 UTC

R topics documented:

`PNAR-package` *Poisson Network Autoregressive Models*

Description

Quasi likelihood-based methods for estimating Poisson Network Autoregression with p lags, PNAR, following generalized linear models are provided. PNAR models with the identity and with the logarithmic link function are allowed. The inclusion of exogenous covariates is also possible. Moreover, it provides tools for testing the linearity of linear PNAR model versus several nonlinear alternatives. Finally, it allows generating multivariate count distributions, from linear and nonlinear PNAR models, where the dependence between Poisson random variables is generated by suitable copulas. References include: Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. `https://arxiv.org/abs/2104.06296` and Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. `https://arxiv.org/abs/2202.03852`

Details

Package:	PNAR
Type:	Package
Version:	1.3
Date:	2022-11-02
License:	GPL(>=2)

Note

Disclaimer: Dr Mirko Armillotta and Konstantinos Fokianos wrote the initial functions. Dr Tsagris modified them, created the package and is the maintainer.

Author(s)

Michail Tsagris, Mirko Armillotta and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

`adja` *Generation of a network from the Stochastic Block Model*

Description

This function generates a network from the Stochastic Block Model with K blocks.

Usage

```
adja(N, K, alpha, directed = FALSE)
```

Arguments

`N`	The number of nodes on the network.
`K`	The number of blocks. Each block has dimension `N/K` so `K` should be chosen such that `N` is divisible by `K`.
`alpha`	The network density. A value in $[0, 1]$ defining the frequency of connections in the network.
`directed`	Logical scalar, whether to generate a directed network or not. If TRUE a directed network is generated.

Details

For each couple of nodes it performs a Bernoulli trial with values 1 "draw an edge", 0 "otherwise". The probabilities of these trials are bigger if the two nodes are in the same block, lower otherwise, and they are specified based on the number of nodes on the network `N` and network density `alpha`. Probability to draw an edge for a couple of nodes in the same block: $\alpha * N^{-0.3}$. Probability to draw an edge for a couple of nodes in different blocks: $\alpha * N^{-1}$.

Value

A row-normalized non-negative matrix describing the network. The main diagonal entries of the matrix are zeros, all the other entries are non-negative and the sum of elements over the rows equals one.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Faust, K. and S. Wasserman (1992). Blockmodels: Interpretation and evaluation. *Social Networks*, 14, 5-61.

See Also

`adja_gnp`

Examples

```
W <- adja(N = 20, K = 5, alpha = 0.1)
```

`adja_gnp` *Generation of a network from the Erdos-Renyi model*

Description

This function generates a network from the Erdos-Renyi model.

Usage

```
adja_gnp(N, alpha, directed = FALSE)
```

Arguments

`N`	The number of nodes on the network.
`alpha`	The network density. A value in $[0, 1]$ defining the frequency of connections in the network.
`directed`	Logical scalar, whether to generate a directed network. If TRUE a directed network is generated.

Details

For each couple of nodes it performs a Bernoulli trial with values 1 "draw an edge", 0 "otherwise". Each trial has the same probability of having an edge; this is equal to $\alpha * N^{-0.3}$, specified based on the number of nodes on the network `N` and the network density `alpha`.

Value

A row-normalized non-negative matrix describing the network. The main diagonal entries of the matrix are zeros, all the other entries are non-negative and the maximum sum of elements over the rows equals one.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Erdos, P. and A. Renyi (1959). On random graphs. *Publicationes Mathematicae*, 6, 290-297.

See Also

`adja`

Examples

```
W <- adja_gnp(N = 20, alpha= 0.1)
```

`crime` *Chicago crime dataset*

Description

Monthly number of burglaries on the south side of Chicago (552 blocks) during 2010-2015 (72 temporal observations).

Usage

```
crime
```

Format

A matrix with 552 rows and 72 columns.

Source

Clark and Dixon (2021), available at `https://github.com/nick3703/Chicago-Data`.

References

Clark, N. J. and P. M. Dixon (2021). A class of spatially correlated self-exciting statistical models. *Spatial Statistics*, 43, 1-18.

See Also

`crime_W`, `lin_estimnarpq`, `log_lin_estimnarpq`

Examples

```
data(crime)
data(crime_W)
mod1 <- lin_estimnarpq(crime, crime_W, p = 1)
mod2 <- log_lin_estimnarpq(crime, crime_W, p = 1)
```

`crime_W` *Network matrix for Chicago crime dataset*

Description

Non-negative row-normized adjacency matrix describing the network structure between Chicago census blocks.

Usage

```
crime_W
```

Format

A matrix with 552 rows and 552 columns.

Source

Clark and Dixon (2021), available at `https://github.com/nick3703/Chicago-Data`.

References

Clark, N. J. and P. M. Dixon (2021). A class of spatially correlated self-exciting statistical models. *Spatial Statistics*, 43, 1-18.

See Also

`crime`, `lin_estimnarpq`, `log_lin_estimnarpq`

Examples

```
data(crime)
data(crime_W)
mod1 <- lin_estimnarpq(crime, crime_W, p = 1)
mod2 <- log_lin_estimnarpq(crime, crime_W, p = 1)
```

`getN` *Count the number of events within a specified time*

Description

This function counts the number of events within a specified time.

Usage

```
getN(x, tt = 1)
```

Arguments

`x`	A matrix of (positive) inter-event times.
`tt`	A positive time.

Value

The number of events within time `tt` (possibly 0), for each column of `x`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

See Also

`rcopula`, `poisson.MODpq`, `poisson.MODpq.log`

Examples

```
x <- rcopula(n = 100, N = 50, rho = 0.3)
getN(x)
```

`global_optimise_LM_stnarpq` *Optimization of the score test statistic for the ST-PNAR(p) model*

Description

Global optimization of the linearity test statistic for the Smooth Transition Poisson Network Autoregressive model of order p with q covariates (ST-PNAR(p)) with respect to the nuisance scale parameter γ.

Usage

```
global_optimise_LM_stnarpq(gama_L = NULL, gama_U = NULL, len = 10, b, y, W,
p, d, Z = NULL, tol = 1e-9)
```

Arguments

`gama_L`	The lower value of the γ values to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`gama_U`	The upper value of the γ values to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`len`	The number of increments to consider for the γ parameter.
`b`	The estimated parameters from the linear model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`tol`	The tolerance level for the optimizer.

Details

The function optimizes the quasi score test statistic, under the null assumption of linearity, for testing linearity of Poisson Network Autoregressive model of order p against the following ST-PNAR(p) model, with respect to the unknown nuisance parameter (γ). For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \alpha_h e^{-\gamma X_{i,t-d}^2} X_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The sequence λ_t is the expectation of Y_t conditional to its past values.

The null hypothesis of the test is defined as $H_0 : \alpha_1 = ... = \alpha_p = 0$, versus the alternative that at least one among α_h is not 0. The test statistic has the form

$$LM(\gamma) = S^{'}(\hat{\theta}, \gamma) \Sigma^{-1}(\hat{\theta}, \gamma) S(\hat{\theta}, \gamma)$$

where

$$S(\hat{\theta}, \gamma) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left(\frac{Y_{i,t}}{\lambda_{i,t}(\hat{\theta}, \gamma)} - 1 \right) \frac{\partial \lambda_{i,t}(\hat{\theta}, \gamma)}{\partial \alpha}$$

is the partition of the quasi score related to the vector of non-linear parameters $\alpha = (\alpha_1, ..., \alpha_p)$, evaluated at the estimated parameters $\hat{\theta}$ under the null assumption H_0 (linear model) and $\Sigma(\hat{\theta}, \gamma)$ is the variance of $S(\hat{\theta}, \gamma)$.

The optimization employes the Brent algorithm applied in the interval from `gama_L` to `gama_U`. To be sure that the global optimum is found, the optimization is performed at (`len`-1) consecutive equidistant sub-intervals and then the maximum over them is taken as global optimum.

The values of `gama_L` and `gama_U` are computed internally as `gama_L` $= -\frac{\log(0.9)}{X^2}$ and `gama_U` $= -\frac{\log(0.1)}{X^2}$, where X is the overall mean of $X_{i,t}$ over the nodes $i = 1, ..., N$ and times $t = 1, ..., TT$. Since the non-linear function $e^{-\gamma X^2_{i,t-d}}$ ranges between 0 and 1, by considering X to be a representative value for the network mean, `gama_U` and `gama_L` would be the values of γ leading the non-linear switching function to be 0.1 and 0.9, respectively, so that in the optimization procedure the extremes of the function domain are excluded. Alternatively, their value can be supplied by the user. For details see Armillotta and Fokianos (2022b, Sec. 4-5).

Value

A list including:

`gama`	The optimum value of the γ parameter.
`supLM`	The value of the objective function at the optimum.
`int`	A vector with the extremes points of sub-intervals.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

See Also

`score_test_stnarpq_j`, `global_optimise_LM_tnarpq`,`score_test_tnarpq_j`

Examples

```
data(crime)
data(crime_W)
mod1 <- lin_estimnarpq(crime, crime_W, p = 1)
b <- mod1$coeflin
global_optimise_LM_stnarpq(b = b, y = crime, W = crime_W, p = 1, d = 1)
```

---

global_optimise_LM_tnarpq

Optimization of the score test statistic for the T-PNAR(p) model

---

Description

Global optimization of the linearity test statistic for the Threshold Poisson Network Autoregressive model of order p with q covariates (T-PNAR(p)) with respect to the nuisance threshold parameter γ.

Usage

```
global_optimise_LM_tnarpq(gama_L = NULL, gama_U = NULL, len = 10, b, y, W,
p, d, Z = NULL, tol = 1e-9)
```

Arguments

`gama_L`	The lower value of the γ values to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`gama_U`	The upper value of the γ values to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`len`	The number of increments to consider for the γ parameter.
`b`	The estimated parameters from the linear model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`tol`	The tolerance level for the optimizer.

Details

The function optimizes the quasi score test statistic, under the null assumption of linearity, for testing linearity of Poisson Network Autoregressive model of order p against the following T-PNAR(p) model, with respect to the unknown nuisance parameter (γ). For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + (\alpha_0 + \alpha_{1h} X_{i,t-h} + \alpha_{2h} Y_{i,t-h}) I(X_{i,t-d} \leq \gamma) + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W, and $I()$ is the indicator function. The sequence λ_t is the expectation of Y_t conditional to its past values.

The null hypothesis of the test is defined as $H_0 : \alpha_0 = \alpha_{11} = ... = \alpha_{2p} = 0$, versus the alternative that at least one among $\alpha_{s,h}$ is not 0, for $s = 0, 1, 2$. The test statistic has the form

$$LM(\gamma) = S^{'}(\hat{\theta}, \gamma) \Sigma^{-1}(\hat{\theta}, \gamma) S(\hat{\theta}, \gamma)$$

where

$$S(\hat{\theta}, \gamma) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left(\frac{Y_{i,t}}{\lambda_{i,t}(\hat{\theta}, \gamma)} - 1 \right) \frac{\partial \lambda_{i,t}(\hat{\theta}, \gamma)}{\partial \alpha}$$

is the partition of the quasi score related to the vector of non-linear parameters $\alpha = (\alpha_0, ..., \alpha_{2p})$, evaluated at the estimated parameters $\hat{\theta}$ under the null assumption H_0 (linear model) and $\Sigma(\hat{\theta}, \gamma)$ is the variance of $S(\hat{\theta}, \gamma)$.

The optimization employes the Brent algorithm applied in the interval from `gama_L` to `gama_U`. To be sure that the global optimum is found, the optimization is performed at (`len`-1) consecutive equidistant sub-intervals and then the maximum over them is taken as global optimum.

The values of `gama_L` and `gama_U` are computed internally as the mean over $i = 1, ..., N$ of 20% and 80% quantile of the empirical distribution of the network mean $X_{i,t}$ for $t = 1, ..., TT$. In this way the optimization is performed for values of γ such that the indicator function $I(X_{i,t-d} \leq \gamma)$ is not always close to 0 or 1. Alternatively, their value can be supplied by the user. For details see Armillotta and Fokianos (2022b, Sec. 4-5).

Value

A list including:

`gama`	The optimum value of the γ parameter.
`supLM`	The value of the objective function at the optimum.
`int`	A vector with the extremes points of sub-intervals.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

See Also

`score_test_tnarpq_j`, `global_optimise_LM_stnarpq`,`score_test_stnarpq_j`

Examples

```
data(crime)
data(crime_W)
mod1 <- lin_estimnarpq(crime, crime_W, p = 2)
b <- mod1$coeflin
global_optimise_LM_tnarpq(b = b, y = crime, W = crime_W, p = 2, d = 1)
```

`lin_estimnarpq` *Estimation of the linear Poisson NAR(p) model model with p lags and q covariates (PNAR(p))*

Description

Estimation of the linear Poisson Network Autoregressive model of order p with q covariates (PNAR(p)).

Usage

```
lin_estimnarpq(y, W, p, Z = NULL, uncons = FALSE)
```

Arguments

`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`uncons`	logical, if TRUE an unconstrained optimization without stationarity constraints is performed (default is FALSE).

Details

This function performs a constrained estimation of the linear Poisson NAR(p) model with q non-negative valued covariates, for each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$, defined as

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The sequence $\lambda_{i,t}$ is the expecation of $Y_{i,t}$ conditional to its past values. The parameter β_0 is the intercept of the model, β_{1h} are the network coefficients, β_{2h} are the autoregressive parameters, and δ_l are the coefficients assocciated to the covariates $Z_{i,l}$.

The estimation of the parameters of the model is performed by Quasi Maximum Likelihood Estimation (QMLE), maximizing the following quasi log-likelihood

$$l(\theta) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left[Y_{i,t} \log \lambda_{i,t}(\theta) - \lambda_{i,t}(\theta) \right]$$

with respect to the vector of unknown parameters θ described above. The coefficients are defined only in the non-negative real line. By default, the optimization is constrained in the stationary region where $\sum_{h=1}^{p} \beta_{1h} + \beta_{2h} < 1$; this can be removed by setting `uncons = TRUE`. However, the user should be warned that the model estimates can be inconsistent if the estimated parameters lie outside the stationary region. The ordinary least squares estimates are employed as starting values of the optimization procedure. Robust standard errors and z-tests are also returned.

Value

A list including:

`coeflin`	A matrix with the estimated QMLE coefficients, their standard errors their Z-test statistics and the relevant p-values computed via the standard normal approximation.
`score`	The value of the quasi score function at the optimization point. It should be close to 0 if the optimization is successful.
`loglik`	The value of the maximized quasi log-likelihood.
`ic`	A vector with the Akaike information criterion (AIC), the Bayesian information criterion (BIC) and the Quasi information criterion (QIC).

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022). Poisson network autoregression. `https://arxiv.org/abs/2104.06296`

See Also

`log_lin_estimnarpq`

Examples

```
data(crime)
data(crime_W)
mod1 <- lin_estimnarpq(crime, crime_W, p = 2)
```

`lin_ic_plot` *Scatter plot of information criteria versus the number of lags in the linear Poisson NAR(p) model model with p lags and q covariates (PNAR(p))*

Description

Scatter plot of information criteria versus the number of lags in the linear Poisson Network Autoregressive model of order p with q covariates (PNAR(p)).

Usage

```
lin_ic_plot(y, W, p = 1:10, Z = NULL, uncons = FALSE)
```

Arguments

`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	A vector with integer numbers, the range of lags in the model, for which the AIC, BIC and QIC will be computed.
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`uncons`	logical, if TRUE an unconstrained optimization without stationarity constraints is performed (default is FALSE).

Details

The function computes the AIC, BIC or QIC for a range of lag orders of the linear Poisson Network Autoregressive model of order p with q covariates (PNAR(p)).

Value

Three scatter plots with the lag order versus AIC, BIC and QIC and a matrix with three columns, the AIC, BIC and QIC for each lag order.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022). Poisson network autoregression. https://arxiv.org/abs/2104.06296

See Also

`lin_estimnarpq`, `log_lin_ic_plot`

Examples

```
data(crime)
data(crime_W)
lin_ic_plot(crime, crime_W, p = 1:3)
```

`log_lin_estimnarpq` *Estimation of the log-linear Poisson NAR(p) model with p lags and q covariates (log-PNAR(p))*

Description

Estimation of the log-linear Poisson Network Autoregressive model of order p with q covariates (log-PNAR(p)).

Usage

```
log_lin_estimnarpq(y, W, p, Z = NULL, uncons = FALSE)
```

Arguments

`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.

`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model.
`uncons`	logical, if TRUE an unconstrained optimization without stationarity constraints is performed (default is FALSE).

Details

This function performs a constrained estimation of the linear Poisson NAR(p) model with q non-negative valued covariates, for each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$, defined as

$$\nu_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The sequence $\nu_{i,t}$ is the log of the expectation of $Y_{i,t}$ conditional to its past values. The parameter β_0 is the intercept of the model, β_{1h} are the network coefficients, β_{2h} are the autoregressive parameters, and δ_l are the coefficients assocciated to the covariates $Z_{i,l}$.

The estimation of the parameters of the model is performed by Quasi Maximum Likelihood Estimation (QMLE), maximizing the following quasi log-likelihood

$$l(\theta) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left[Y_{i,t} \nu_{i,t}(\theta) - e^{\nu_{i,t}(\theta)} \right]$$

with respect to the vector of unknown parameters θ described above. By default, the optimization is constrained in the stationary region where $\sum_{h=1}^{p} |\beta_{1h}| + |\beta_{2h}| < 1$; this can be removed by setting `uncons = TRUE`. However, the user should be warned that the model estimates can be inconsistent if the estimated parameters lie outside the stationary region. The ordinary least squares estimates are employed as starting values of the optimization procedure. Robust standard errors and z-tests are also returned.

Value

A list including:

`coeflog`	A matrix with the estimated QMLE coefficients, their standard errors, their Z-test statistics and the relevant p-values computed via the standard normal approximation.
`score`	The value of the quasi score function at the optimization point. It should be close to 0 if the optimization is successful.
`loglik`	The value of the maximized quasi log-likelihood.
`ic`	A vector with the Akaike information criterion (AIC), the Bayesian information criterion (BIC) and the Quasi information criterion (QIC).

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022). Poisson network autoregression. https://arxiv.org/abs/2104.06296

See Also

`lin_estimnarpq`

Examples

```
data(crime)
data(crime_W)
mod1 <- log_lin_estimnarpq(crime, crime_W, p = 2)
```

`log_lin_ic_plot` *Scatter plot of information criteria versus the number of lags in the log-linear Poisson NAR(p) model with p lags and q covariates (log-PNAR(p))*

Description

Scatter plot of information criteria versus the number of lags in log-linear Poisson Network Autoregressive model of order p with q covariates (log-PNAR(p)).

Usage

```
log_lin_ic_plot(y, W, p = 1:10, Z = NULL, uncons = FALSE)
```

Arguments

`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	A vector with integer numbers, the range of lags in the model, for which the AIC, BIC and QIC will be computed.
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`uncons`	logical, if TRUE an unconstrained optimization without stationarity constraints is performed (default is FALSE).

Details

The function computes the AIC, BIC or QIC for a range of lag orders of the linear Poisson Network Autoregressive model of order p with q covariates (PNAR(p)).

Value

Three scatter plots with the lag order versus AIC, BIC and QIC and a matrix with three columns, the AIC, BIC and QIC for each lag order.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022). Poisson network autoregression. `https://arxiv.org/abs/2104.06296`

See Also

`log_lin_estimnarpq`, `lin_ic_plot`

Examples

```
data(crime)
data(crime_W)
log_lin_ic_plot(crime, crime_W, p = 1:3)
```

`poisson.MODpq` *Generation of counts from a linear Poisson NAR(p) model with q covariates (PNAR(p))*

Description

Generation of counts from a linear Poisson Network Autoregressive model of order p with q covariates (PNAR(p)).

Usage

```
poisson.MODpq(b, W, p, Z = NULL, TT, N, copula = "gaussian",
corrtype = "equicorrelation", rho, dof = 1)
```

Arguments

`b`	The coefficients of the model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2`p` + 1 + number of covariates.
`W`	The `N x N` row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`Z`	An `N` x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`TT`	The temporal sample size.
`N`	The number of nodes on the network.
`copula`	Which copula function to use? The "gaussian", "t", or "clayton"?
`rho`	The value of the copula parameter (ρ). A scalar in $[-1, 1]$ for elliptical copulas (Gaussian, t), a value greater than or equal to -1 for Clayton copula.
`corrtype`	Used only for elliptical copulas. The type of correlation matrix employed for the copula; it will either be the "equicorrelation" or "toeplitz". The "equicorrelation" option generates a correlation matrix where all the off-diagonal entries equal ρ. The "toeplitz" option generates a correlation matrix whose generic off-diagonal (i, j)-element is $\rho^{\lvert i-j \rvert}$.
`dof`	The degrees of freedom for Student's t copula.

Details

This function generates counts from a linear Poisson NAR(p) model, where q non time-varying covariates are allowed as well. The counts are simulated from $Y_t = N_t(\lambda_t)$, where N_t is a sequence of `N`-dimensional IID Poisson count processes, with intensity 1, and whose structure of dependence is modelled through a copula construction $C(\rho)$ on their associated exponential waiting times random variables. For details see Armillotta and Fokianos (2022, Sec. 2.1-2.2).

The sequence λ_t is the expectation of Y_t conditional to its past values and it is generated by means of the following PNAR(p) model. For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The parameter β_0 is the intercept of the model, β_{1h} are the network coefficients, β_{2h} are the autoregressive parameters, and δ_l are the coefficients assocciated to the covariates $Z_{i,l}$.

Value

A list including:

`p2R`	The Toeplitz correlation matrix, if employed in the copula or NULL else.
`lambda`	An `N` x `TT` matrix of generated Poisson means for `N` time series over `TT`.
`y`	An `N` x `TT` matrix of generated counts for `N` time series over `TT`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Fokianos, K., Stove, B., Tjostheim, D., and P. Doukhan (2020). Multivariate count autoregression. Bernoulli, 26(1), 471-499.

See Also

`poisson.MODpq.log`, `poisson.MODpq.nonlin`,`poisson.MODpq.stnar`, `poisson.MODpq.tnar`

Examples

```
W <- adja( N = 20, K = 5, alpha= 0.5)
y <- poisson.MODpq( b = c(0.5, 0.3, 0.2), W = W, p = 1, Z = NULL,
TT = 1000, N = 20, copula = "gaussian",
corrtype = "equicorrelation", rho = 0.5)$y
```

`poisson.MODpq.log` *Generation of counts from a log-linear Poisson NAR(p) model with q covariates (log-PNAR(p))*

Description

Generation of counts from a log-linear Poisson Network Autoregressive model of order p with q covariates (log-PNAR(p)).

Usage

```
poisson.MODpq.log(b, W, p, Z = NULL, TT, N, copula = "gaussian",
corrtype = "equicorrelation", rho, dof = 1)
```

Arguments

`b`	The coefficients of the model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.

`W`	The `N x N` row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one.The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`Z`	An `N` x q matrix of covariates (one for each column), where q is the number of covariates in the model.
`TT`	The temporal sample size.
`N`	The number of nodes on the network.
`copula`	Which copula function to use? The "gaussian", "t", or "clayton"?
`rho`	The the value of the copula parameter (ρ). A scalar in $[-1, 1]$ for elliptical copulas (Gaussian, t), a value greater or equal to -1 for Clayton copula.
`corrtype`	Used only for elliptical copulas. The type of correlation matrix employed for the copula; it will either be the "equicorrelation" or "toeplitz". The "equicorrelation" option generates a correlation matrix where all the off-diagonal entries equal ρ. The "toeplitz" option generates a correlation matrix whose generic off-diagonal (i, j)-element is $\rho^{\|i-j\|}$.
`dof`	The degrees of freedom for Student's t copula.

Details

This function generates counts from a log-linear Poisson NAR(p) model, where q non time-varying covariates are allowed as well. The counts are simulated from $Y_t = N_t(e^{\nu_t})$, where N_t is a sequence of `N`-dimensional IID Poisson count processes, with intensity 1, and whose structure of dependence is modelled through a copula construction $C(\rho)$ on their associated exponential waiting times random variables. For details see Armillotta and Fokianos (2022, Sec. 2.1-2.2).

The sequence ν_t is the log of the expecation of Y_t conditional to its past values and it is generated by means of the following log-PNAR(p) model. For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\nu_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The parameter β_0 is the intercept of the model, β_{1h} are the network coefficients, β_{2h} are the autoregressive parameters, and δ_l are the coefficients assocciated to the covariates $Z_{i,l}$.

Value

A list including:

`p2R`	The Toeplitz correlation matrix, if employed in the copula or NULL else.
`log_lambda`	An `N` x `TT` matrix of generated Poisson log-means for `N` time series over `TT`.
`y`	An `N` x `TT` matrix of generated counts for `N` time series over `TT`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Fokianos, K., Stove, B., Tjostheim, D., and P. Doukhan (2020). Multivariate count autoregression. Bernoulli, 26(1), 471-499.

See Also

`poisson.MODpq`, `poisson.MODpq.nonlin`,`poisson.MODpq.stnar`, `poisson.MODpq.tnar`

Examples

```
W <- adja( N = 20, K = 5, alpha= 0.5)
y <- poisson.MODpq.log( b = c(0.5, 0.3, 0.2), W = W, p = 1,
Z = NULL, TT = 1000, N = 20, copula = "gaussian",
corrtype = "equicorrelation", rho = 0.5)$y
```

`poisson.MODpq.nonlin` *Generation of counts from a non-linear Intercept Drift Poisson NAR(p) model with q covariates (ID-PNAR(p))*

Description

Generation of counts from a non-linear Intercept Drift Poisson Network Autoregressive model of order p with q covariates (ID-PNAR(p)).

Usage

```
poisson.MODpq.nonlin(b, W, gama, p, d, Z = NULL, TT, N,
copula = "gaussian", corrtype = "equicorrelation", rho, dof = 1)
```

Arguments

`b`	The linear coefficients of the model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`W`	The `N x N` row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`gama`	A scalar non-linear intercept drift parameter.

`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An `N` x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`TT`	The temporal sample size.
`N`	The number of nodes on the network.
`copula`	Which copula function to use? The "gaussian", "t", or "clayton"?
`rho`	The value of the copula parameter (ρ). A scalar in $[-1, 1]$ for elliptical copulas (Gaussian, t), a value greater than or equal to -1 for Clayton copula.
`corrtype`	Used only for elliptical copulas. The type of correlation matrix employed for the copula; it will either be the "equicorrelation" or "toeplitz". The "equicorrelation" option generates a correlation matrix where all the off-diagonal entries equal ρ. The "toeplitz" option generates a correlation matrix whose generic off-diagonal (i, j)-element is $\rho^{\lvert i-j \rvert}$.
`dof`	The degrees of freedom for Student's t copula.

Details

This function generates counts from a non-linear Intercept Drift Poisson NAR(p) model, where q non time-varying covariates are allowed as well. The counts are simulated from $Y_t = N_t(\lambda_t)$, where N_t is a sequence of `N`-dimensional IID Poisson count processes, with intensity 1, and whose structure of dependence is modelled through a copula construction $C(\rho)$ on their associated exponential waiting times random variables. For details see Armillotta and Fokianos (2022a, Sec. 2.1-2.2). The sequence λ_t is the expecation of Y_t conditional to its past values and it is generated by means of the following ID-PNAR(p) model. For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \frac{\beta_0}{(1 + X_{i,t-d})^{\gamma}} + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W.

The parameter β_0 is the intercept of the model, β_{1h} are the network coefficients, β_{2h} are the autoregressive parameters, γ is the non-linear coefficient associated with the intercept drift, and δ_l are the coefficients assocciated with the covariates $Z_{i,l}$. The coefficient d is considered as an extra parameter defining the lag of the network effect in the non-linear part of the model and is left to be set by the user. For details on ID-PNAR models see Armillotta and Fokianos (2022b, Sec. 2).

Value

A list including:

`p2R`	The Toeplitz correlation matrix, if employed in the copula or NULL else.
`lambda`	An `N` x `TT` matrix of generated Poisson means for `N` time series over `TT`.
`y`	An `N` x `TT` matrix of generated counts for `N` time series over `TT`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

See Also

`poisson.MODpq`, `poisson.MODpq.log`,`poisson.MODpq.stnar`, `poisson.MODpq.tnar`

Examples

```
W <- adja( N = 20, K = 5, alpha= 0.5)
y <- poisson.MODpq.nonlin( b = c(0.5, 0.3, 0.2), W = W, gama = 1, p = 1,
d = 1, Z = NULL, TT = 1000, N = 20, copula = "gaussian",
corrtype = "equicorrelation", rho = 0.5)$y
```

`poisson.MODpq.stnar` *Generation of counts from a non-linear Smooth Transition Poisson NAR(p) model with q covariates (ST-PNAR(p))*

Description

Generation of counts from a non-linear Smooth Transition Poisson Network Autoregressive model of order p with q covariates (ST-PNAR(p)).

Usage

```
poisson.MODpq.stnar(b, W, gama, a, p, d, Z = NULL, TT, N,
copula = "gaussian", corrtype = "equicorrelation", rho, dof = 1)
```

Arguments

`b`	The linear coefficients of the model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`W`	The `N x N` row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`gama`	The scalar nuisance smoothing parameter.

`a`	Vector of non-linear parameters. The dimension of the vector should be `p`.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An `N` x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`TT`	The temporal sample size.
`N`	The number of nodes on the network.
`copula`	Which copula function to use? The "gaussian", "t", or "clayton"?
`rho`	The value of the copula parameter (ρ). A scalar in $[-1, 1]$ for elliptical copulas (Gaussian, t), a value greater than or equal to -1 for Clayton copula.
`corrtype`	Used only for elliptical copulas. The type of correlation matrix employed for the copula; it will either be the "equicorrelation" or "toeplitz". The "equicorrelation" option generates a correlation matrix where all the off-diagonal entries equal ρ. The "toeplitz" option generates a correlation matrix whose generic off-diagonal (i, j)-element is $\rho^{\lvert i-j \rvert}$.
`dof`	The degrees of freedom for Student's t copula.

Details

This function generates counts from a non-linear Smooth Transition Poisson NAR(p) model, where q non time-varying covariates are allowed as well. The counts are simulated from $Y_t = N_t(\lambda_t)$, where N_t is a sequence of `N`-dimensional IID Poisson count processes, with intensity 1, and whose structure of dependence is modelled through a copula construction $C(\rho)$ on their associated exponential waiting times random variables. For details see Armillotta and Fokianos (2022a, Sec. 2.1-2.2).

The sequence λ_t is the expecation of Y_t conditional to its past values and it is generated by means of the following ST-PNAR(p) model. For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \alpha_h e^{-\gamma X_{i,t-d}^2} X_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The parameter β_0 is the intercept of the model, β_{1h} are the network coefficients, β_{2h} are the autoregressive parameters, α_h are the non-linear smooth transition parameters, γ is the nuisance smoothing parameter, and δ_l are the coefficients assocciated to the covariates $Z_{i,l}$. The coefficient d is considered as an extra parameter defining the lag of the network effect in the non-linear part of the model and is left to be set by the user. For details on ST-PNAR models see Armillotta and Fokianos (2022b, Sec. 2).

Value

A list including:

`p2R`	The Toeplitz correlation matrix, if employed in the copula or NULL else.
`lambda`	An `N` x `TT` matrix of generated Poisson means for `N` time series over `TT`.
`y`	An `N` x `TT` matrix of generated counts for `N` time series over `TT`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

See Also

`poisson.MODpq`, `poisson.MODpq.log`,`poisson.MODpq.nonlin`, `poisson.MODpq.tnar`

Examples

```
W <- adja( N = 20, K = 5, alpha= 0.5)
y <- poisson.MODpq.stnar( b = c(0.5, 0.3, 0.2), W = W, gama = 0.2, a = 0.4,
p = 1, d = 1, Z = NULL, TT = 1000, N = 20, copula = "gaussian",
corrtype = "equicorrelation", rho = 0.5)$y
```

`poisson.MODpq.tnar` *Generation of counts from a non-linear Threshold Poisson NAR(p) model with q covariates (T-PNAR(p))*

Description

Generation of counts from a non-linear Threshold Poisson network Autoregressive model of order p with q covariates (T-PNAR(p)).

Usage

```
poisson.MODpq.tnar(b, W, gama, a, p, d, Z = NULL, TT, N,
copula = "gaussian", corrtype = "equicorrelation", rho, dof = 1)
```

Arguments

`b`	The linear coefficients of the model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`W`	The `N x N` row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`gama`	The scalar nuisance threshold parameter.

`a`	Vector of non-linear parameters. The dimension of the vector should be 2p + 1.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An `N` x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`TT`	The temporal sample size.
`N`	The number of nodes on the network.
`copula`	Which copula function to use? The "gaussian", "t", or "clayton"?
`rho`	The value of the copula parameter (ρ). A scalar in $[-1, 1]$ for elliptical copulas (Gaussian, t), a value greater than or equal to -1 for Clayton copula.
`corrtype`	Used only for elliptical copulas. The type of correlation matrix employed for the copula; it will either be the "equicorrelation" or "toeplitz". The "equicorrelation" option generates a correlation matrix where all the off-diagonal entries equal ρ. The "toeplitz" option generates a correlation matrix whose generic off-diagonal (i, j)-element is $\rho^{\|i-j\|}$.
`dof`	The degrees of freedom for Student's t copula.

Details

This function generates counts from a non-linear Threshold Poisson NAR(p) model, where q non time-varying covariates are allowed as well. The counts are simulated from $Y_t = N_t(\lambda_t)$, where N_t is a sequence of `N`-dimensional IID Poisson count processes, with intensity 1, and whose structure of dependence is modelled through a copula construction $C(\rho)$ on their associated exponential waiting times random variables. For details see Armillotta and Fokianos (2022a, Sec. 2.1-2.2).

The sequence λ_t is the expecation of Y_t conditional to its past values and it is generated by means of the following T-PNAR(p) model. For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + (\alpha_0 + \alpha_{1h} X_{i,t-h} + \alpha_{2h} Y_{i,t-h}) I(X_{i,t-d} \leq \gamma) + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W, and $I()$ is the indicator function.

The parameter β_0 is the intercept of the model, β_{1h} are the network coefficients, β_{2h} are the autoregressive parameters, the α vector of non-linear parameters is divided as follows: α_0 is the intercept, α_{1h} are the network coefficients, α_{2h} are the autoregressive parameters; γ is the nuisance threshold parameter, and δ_l are the coefficients assocciated to the covariates $Z_{i,l}$. The coefficient d is considered as an extra parameter defining the lag of the network effect in the non-linear part of the model and is left to be set by the user. For details on T-PNAR models see Armillotta and Fokianos (2022b, Sec. 2).

Value

A list including:

`p2R`	The Toeplitz correlation matrix, if employed in the copula or NULL else.
`lambda`	An `N` x `TT` matrix of generated Poisson means for `N` time series over `TT`.
`y`	An `N` x `TT` matrix of generated counts for `N` time series over `TT`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. `https://arxiv.org/abs/2104.06296`

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. `https://arxiv.org/abs/2202.03852`

See Also

`poisson.MODpq`, `poisson.MODpq.log`,`poisson.MODpq.nonlin`, `poisson.MODpq.stnar`

Examples

```
W <- adja( N = 20, K = 5, alpha= 0.5)
y <- poisson.MODpq.tnar( b = c(0.5, 0.3, 0.2), W = W, gama = 1,
a = c(0.2, 0.2, 0.2), p = 1, d = 1, Z = NULL, TT = 1000, N = 20,
copula = "gaussian", corrtype = "equicorrelation", rho = 0.5)$y
```

`rcopula` *Random number generation of copula functions*

Description

Random number generation of copula functions.

Usage

```
rcopula(n, N, copula = "gaussian", corrtype = "equicorrelation",
rho, dof = 1, cholR = NULL)
```

Arguments

`n`	The number of random values to generate.
`N`	The number of variables for which random valeus will be generated.
`copula`	Which copula function to use? The "gaussian", "t", or "clayton"?
`rho`	The the value of the copula parameter (ρ). A scalar in $[-1, 1]$ for elliptical copulas (Gaussian, t), a value greater than or equal to -1 for Clayton copula.

`corrtype`	Used only for elliptical copulas. The type of correlation matrix employed for the copula; it will either be the "equicorrelation" or "toeplitz". The "equicorrelation" option generates a correlation matrix where all the off-diagonal entries equal ρ. The "toeplitz" option generates a correlation matrix whose generic off-diagonal (i, j)-element is $\rho^{\lvert i-j \rvert}$.
`dof`	The degrees of freedom for Student's t copula.
`cholR`	An alternative input for elliptic copulas, providing directly the Cholesky decomposition for a specific correlation matrix to be passed, otherwise leave it NULL.

Details

This function generates random copula values from Gaussian, Student's t, or Clayton copulas based on a single copula paremeter and different correlation structures.

Value

An `n` x `N` matrix with the simulated copula values.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Nelsen, Roger B. (1999). *An Introduction to Copulas*, Springer.

See Also

`getN`, `poisson.MODpq`, `poisson.MODpq.log`

Examples

```
u <- rcopula(n = 100, N = 50, rho = 0.3)
```

`score_test_nonlinpq_h0`

Linearity test against non-linear ID-PNAR(p) model

Description

Quasi score test for testing linearity of Poisson Network Autoregressive model of order p against the non-linear Intercep Drift (ID) version (ID-PNAR(p)).

Usage

```
score_test_nonlinpq_h0(b, y, W, p, d, Z = NULL)
```

Arguments

`b`	The estimated parameters from the linear PNAR model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.

Details

The function computes the quasi score test for testing linearity of Poisson Network Autoregressive model of order p against the following ID-PNAR(p) model. For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \frac{\beta_0}{(1 + X_{i,t-d})^{\gamma}} + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The sequence λ_t is the expectation of Y_t conditional to its past values.

The null hypothesis of the test is defined as $H_0 : \gamma = 0$, versus the alternative $H_1 : \gamma > 0$. The test statistic has the form

$$LM = S^{'}(\hat{\theta}) \Sigma^{-1}(\hat{\theta}) S(\hat{\theta})$$

where

$$S(\hat{\theta}) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left(\frac{Y_{i,t}}{\lambda_{i,t}(\hat{\theta})} - 1 \right) \frac{\partial \lambda_{i,t}(\hat{\theta})}{\partial \gamma}$$

is the partition of the quasi score related to the non-linear parameter γ, evaluated at the estimated parameters $\hat{\theta}$ under the null assumption H_0 (linear model), and $\Sigma(\hat{\theta})$ is the variance of $S(\hat{\theta})$. Under H_0, the test asymptotically follows a chi-square with 1 degree of freedom. For details see Armillotta and Fokianos (2022b, Sec. 4).

Value

A list including:

`stat`	The value of the chi-square test statistic.
`pvalue`	The p-value of the chi-square test statistic.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

See Also

`score_test_stnarpq_j`, `score_test_tnarpq_j`,`lin_estimnarpq`

Examples

```
data(crime)
data(crime_W)
mod1 <- lin_estimnarpq(crime, crime_W, p = 2)
ca <- mod1$coeflin[, 1]
score_test_nonlinpq_h0(ca, crime, crime_W, p = 2, d = 1)
```

`score_test_stnarpq_DV` *Bound p-value for testing for smooth transition effects on PNAR(p) model*

Description

Computation of Davies bound p-value for the sup-type test for testing linearity of Poisson Network Autoregressive model of order p (PNAR(p)) versus the non-linear Smooth Transition alternative (ST-PNAR(p)).

Usage

```
score_test_stnarpq_DV(b, y, W, p, d, Z = NULL, gama_L = NULL,
gama_U = NULL, len = 100)
```

Arguments

`b`	The estimated parameters from the linear model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.

`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`gama_L`	The lower value of the nuisance parameter γ to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`gama_U`	The upper value of the nuisance parameter γ to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`len`	The length of the grid of values of γ values to consider.

Details

The function computes an upper-bound for the p-value of the sup-type test for testing linearity of Poisson Network Autoregressive model of order p (PNAR(p)) versus the following Smooth Transition alternative (ST-PNAR(p)). For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \alpha_h e^{-\gamma X_{i,t-d}^2} X_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The sequence λ_t is the expectation of Y_t conditional to its past values.

The null hypothesis of the test is defined as $H_0 : \alpha_1 = ... = \alpha_p = 0$, versus the alternative that at least one among α_h is not 0. The test statistic has the form

$$LM(\gamma) = S^{'}(\hat{\theta}, \gamma) \Sigma^{-1}(\hat{\theta}, \gamma) S(\hat{\theta}, \gamma),$$

where

$$S(\hat{\theta}, \gamma) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left(\frac{Y_{i,t}}{\lambda_{i,t}(\hat{\theta}, \gamma)} - 1 \right) \frac{\partial \lambda_{i,t}(\hat{\theta}, \gamma)}{\partial \alpha}$$

is the partition of the quasi score related to the vector of non-linear parameters $\alpha = (\alpha_1, ..., \alpha_p)$, evaluated at the estimated parameters $\hat{\theta}$ under the null assumption H_0 (linear model), and $\Sigma(\hat{\theta}, \gamma)$ is the variance of $S(\hat{\theta}, \gamma)$. Since the test statistic depends on an unknown nuisance parameter (γ), the supremum of the statistic is considered in the test, $\sup_\gamma LM(\gamma)$. The function computes the bound of the p-value, suggested by Davies (1987), for the test statistic $\sup_\gamma LM(\gamma)$, with scalar nuisance parameter γ, as follows. $P(\chi_k^2 \geq M) + V M^{1/2(k-1)} \frac{e^{-M/2} 2^{-k/2}}{\Gamma(k/2)}$ where M is the maximum of the test statistic $LM(\gamma)$, computed by the available sample, over a grid of values for the nuisance parameter $\gamma_F = (\gamma_L, \gamma_1, ..., \gamma_l, \gamma_U)$; k is the number of non-linear parameters tested. So the first summand of the bound is just the p-value of a chi-square test with k degrees of freedom. The second summand is a correction term depending on V, which is the approximated total variation computed as $V = |LM^{1/2}(\gamma_1) - LM^{1/2}(\gamma_L)| + |LM^{1/2}(\gamma_2) - LM^{1/2}(\gamma_1)| + ... + |LM^{1/2}(\gamma_U) - LM^{1/2}(\gamma_l)|$. The feasible bound allows to approximate the p-values of the sup-type test in a straightforward way, by adding to the tail probability of a chi-square distribution a correction term which depends on the total variation of the process. For details see Armillotta and Fokianos (2022b, Sec. 5).

The values of `gama_L` and `gama_U` are computed internally as `gama_L` $= -\frac{\log(0.9)}{X^2}$ and `gama_U` $= -\frac{\log(0.1)}{X^2}$, where X is the overall mean of $X_{i,t}$ over the nodes $i = 1, ..., N$ and times $t = 1, ..., TT$. Since the non-linear function $e^{-\gamma X^2_{i,t-d}}$ ranges between 0 and 1, by considering X to be a representative value for the network mean, `gama_U` and `gama_L` would be the values of γ leading the non-linear switching function to be 0.1 and 0.9, respectively, so that in the optimization procedure the extremes of the function domain are excluded. Alternatively, their values can be supplied by the user.

Value

A list including:

`DV`	The Davies bound of p-values for sup test.
`supLM`	The value of the sup test statistic in the sample `y`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. `https://arxiv.org/abs/2104.06296`

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. `https://arxiv.org/abs/2202.03852`

Davies, R. B. (1987). Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika* 74, 33-43.

See Also

`score_test_stnarpq_j`, `global_optimise_LM_stnarpq`

Examples

```
data(crime)
data(crime_W)
mod1 <- lin_estimnarpq(crime, crime_W, p = 1)
ca <- mod1$coeflin[, 1]
score_test_stnarpq_DV(ca, crime, crime_W, p = 1, d = 1)
```

`score_test_stnarpq_j` *Bootstrap test for smooth transition effects on PNAR(p) model*

Description

Computation of bootstrap p-value for the sup-type test for testing linearity of Poisson Network Autoregressive model of order p (PNAR(p)) versus the non-linear Smooth Transition alternative (ST-PNAR(p)).

Usage

```
score_test_stnarpq_j(supLM, b, y, W, p, d, Z = NULL, J = 499,
gama_L = NULL, gama_U = NULL, tol = 1e-9, ncores = 1, seed = NULL)
```

Arguments

`supLM`	The optimized value of the test statistic. See the function `global_optimise_LM_stnarpq`.
`b`	The estimated parameters from the linear model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.
`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`J`	The number of bootstrap samples to draw.
`gama_L`	The lower value of the nuisance parameter γ to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`gama_U`	The upper value of the nuisance parameter γ to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`tol`	The tolerance level for the optimizer.
`ncores`	The number of cores to use for parallel computing. By default the number of cores is set to 1 (no parallel computing). **Note:** If for some reason the parallel does not work then load the doParallel package yourseleves.
`seed`	If you want the same to be generated again use a seed for the generator, an integer number.

Details

The function computes a bootstrap p-value for the sup-type test for testing linearity of Poisson Network Autoregressive model of order p (PNAR(p)) versus the following Smooth Transition alternative (ST-PNAR(p)). For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + \alpha_h e^{-\gamma X^2_{i,t-d}} X_{i,t-h} + \sum_{l=1}^{q} \delta_l Z_{i,l},$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The sequence λ_t is the expectation of Y_t conditional to its past values.

The null hypothesis of the test is defined as $H_0 : \alpha_1 = ... = \alpha_p = 0$, versus the alternative that at least one among α_h is not 0. The test statistic has the form

$$LM(\gamma) = S^{'}(\hat{\theta}, \gamma) \Sigma^{-1}(\hat{\theta}, \gamma) S(\hat{\theta}, \gamma)$$

where

$$S(\hat{\theta}, \gamma) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left(\frac{Y_{i,t}}{\lambda_{i,t}(\hat{\theta}, \gamma)} - 1 \right) \frac{\partial \lambda_{i,t}(\hat{\theta}, \gamma)}{\partial \alpha}$$

is the partition of the quasi score related to the vector of non-linear parameters $\alpha = (\alpha_1, ..., \alpha_p)$, evaluated at the estimated parameters $\hat{\theta}$ under the null assumption H_0 (linear model), and $\Sigma(\hat{\theta}, \gamma)$ is the variance of $S(\hat{\theta}, \gamma)$.

Since the test statistic depends on an unknown nuisance parameter (γ), the supremum of the statistic is considered in the test, $\sup_{\gamma} LM(\gamma)$. This value can be computed for the available sample by using the function `global_optimise_LM_stnarpq` and should be supplied here as an input `supLM`.

The function performs the bootstrap resampling of the test statistic $\sup_{\gamma} LM(\gamma)$ by employing Gaussian perturbations of the score $S(\hat{\theta}, \gamma)$. For details see Armillotta and Fokianos (2022b, Sec. 5).

The values of `gama_L` and `gama_U` are computed internally as `gama_L` $= -\frac{\log(0.9)}{X^2}$ and `gama_U` $= -\frac{\log(0.1)}{X^2}$, where X is the overall mean of $X_{i,t}$ over the nodes $i = 1, ..., N$ and times $t = 1, ..., TT$. Since the non-linear function $e^{-\gamma X^2_{i,t-d}}$ ranges between 0 and 1, by considering X to be a representative value for the network mean, `gama_U` and `gama_L` would be the values of γ leading the non-linear switching function to be 0.1 and 0.9, respectively, so that in the optimization procedure the extremes of the function domain are excluded. Alternatively, their value can be supplied by the user.

Note: Please note that for large datasets the function may require few minutes to run. Parallel computing is suggested to speed up the computations.

Value

A list including:

`pJ`	The bootstrap p-value of the sup test.
`cpJ`	The adjusted version of bootstrap p-value of the sup test.
`gamaj`	The optimal values of the γ parameter for score test boostrap replications.
`supLMj`	The values of perturbed test statistic at the optimum point `gamaj`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

See Also

`score_test_stnarpq_DV`, `global_optimise_LM_stnarpq`,`score_test_tnarpq_j`

Examples

```
# load data
data(crime)
data(crime_W)
#estimate linear PNAR model
mod1 <- lin_estimnarpq(crime, crime_W, p = 2)
b <- mod1$coeflin[, 1]

g <- global_optimise_LM_stnarpq(b = b, y = crime, W = crime_W, p = 2, d = 1)
supg <- g$supLM
score_test_stnarpq_j(supLM = supg, b = b, y = crime, W = crime_W, p = 2, d = 1, J = 5)
```

`score_test_tnarpq_j` *Bootstrap test for threshold effects on PNAR(p) model*

Description

Computation of bootstrap p-value for the sup-type test for testing linearity of Poisson Network Autoregressive model of order p (PNAR(p)) versus the non-linear Threshold alternative (T-PNAR(p)).

Usage

```
score_test_tnarpq_j(supLM, b, y, W, p, d, Z = NULL, J = 499,
gama_L = NULL, gama_U = NULL, tol = 1e-9, ncores = 1, seed = NULL)
```

Arguments

`supLM`	The optimized value of the test statistic. See the function `global_optimise_LM_tnarpq`.
`b`	The estimated parameters from the linear model, in the following order: (intercept, network parameters, autoregressive parameters, covariates). The dimension of the vector should be 2p + 1 + number of covariates.

`y`	An N x TT numerical matrix with the N multivariate count time series over TT time periods.
`W`	The N x N row-normalized non-negative adjacency matrix describing the network. The main diagonal entries of the matrix should be zeros, all the other entries should be non-negative and the maximum sum of elements over the rows should equal one. The function row-normalizes the matrix if a non-normalized adjacency matrix is provided.
`p`	The number of lags in the model.
`d`	The lag parameter of non-linear variable (should be between 1 and p).
`Z`	An N x q matrix of covariates (one for each column), where q is the number of covariates in the model. Note that they must be non-negative.
`J`	The number of bootstrap samples to draw.
`gama_L`	The lower value of the nuisance parameter γ to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`gama_U`	The upper value of the nuisance parameter γ to consider. If you do not know the value leave it NULL and it will be computed internally. See the details for information about default computation.
`tol`	The tolerance level for the optimizer.
`ncores`	The number of cores to use for parallel computing. By default the number of cores is set to 1 (no parallel computing). **Note:** If for some reason the parallel does not work then load the doParallel package yourseleves.
`seed`	If you want the same to be generated again use a seed for the generator, an integer number.

Details

The function computes a bootstrap p-value for the sup-type test for testing linearity of Poisson Network Autoregressive model of order p (PNAR(p)) versus the following Threshold alternative (T-PNAR(p)). For each node of the network $i = 1, ..., N$ over the time sample $t = 1, ..., TT$

$$\lambda_{i,t} = \beta_0 + \sum_{h=1}^{p} \beta_{1h} X_{i,t-h} + \beta_{2h} Y_{i,t-h} + (\alpha_0 + \alpha_{1h} X_{i,t-h} + \alpha_{2h} Y_{i,t-h}) I(X_{i,t-d} \leq \gamma) + \sum_{l=1}^{q} \delta_l Z_{i,l}$$

where $X_{i,t} = \sum_{j=1}^{N} W_{ij} Y_{j,t}$ is the network effect, i.e. the weighted average impact of node i connections, with the weights of the mean being W_{ij}, the single element of the network matrix W. The sequence λ_t is the expectation of Y_t conditional to its past values.

The null hypothesis of the test is defined as $H_0 : \alpha_0 = \alpha_{11} = ... = \alpha_{2p} = 0$, versus the alternative that at least one among $\alpha_{s,h}$ is not 0, for $s = 0, 1, 2$. The test statistic has the form

$$LM(\gamma) = S^{'}(\hat{\theta}, \gamma) \Sigma^{-1}(\hat{\theta}, \gamma) S(\hat{\theta}, \gamma)$$

where

$$S(\hat{\theta}, \gamma) = \sum_{t=1}^{TT} \sum_{i=1}^{N} \left(\frac{Y_{i,t}}{\lambda_{i,t}(\hat{\theta}, \gamma)} - 1 \right) \frac{\partial \lambda_{i,t}(\hat{\theta}, \gamma)}{\partial \alpha}$$

is the partition of the quasi score related to the vector of non-linear parameters $\alpha = (\alpha_0, ..., \alpha_{2p})$, evaluated at the estimated parameters $\hat{\theta}$ under the null assumption H_0 (linear model), and $\Sigma(\hat{\theta}, \gamma)$ is the variance of $S(\hat{\theta}, \gamma)$.

Since the test statistic depends on an unknown nuisance parameter (γ), the supremum of the statistic is considered in the test, $\sup_\gamma LM(\gamma)$. This value can be computed for the available sample by using the function `global_optimise_LM_tnarpq` and should be supplied here as an input `supLM`.

The function performs the bootstrap resampling of the test statistic $\sup_\gamma LM(\gamma)$ by employing Gaussian perturbations of the score $S(\hat{\theta}, \gamma)$. For details see Armillotta and Fokianos (2022b, Sec. 5).

The values of `gama_L` and `gama_U` are computed internally as the mean over $i = 1, ..., N$ of 20% and 80% quantiles of the empirical distribution of the network mean $X_{i,t}$ for $t = 1, ..., TT$. In this way the optimization is performed for values of γ such that the indicator function $I(X_{i,t-d} \leq \gamma)$ is not always close to 0 or 1. Alternatively, their value can be supplied by the user. For details see Armillotta and Fokianos (2022b, Sec. 4-5).

Note: Please note that for large datasets the function may require few minutes to run. Parallel computing is suggested to speed up the computations.

Value

A list including:

`pJ`	The bootstrap p-value of the sup test.
`cpJ`	The adjusted version of bootstrap p-value of the sup test.
`gamaj`	The optimal values of the γ parameter for score test boostrap replications.
`supLMj`	The values of perturbed test statistic at the optimum point `gamaj`.

Author(s)

Mirko Armillotta, Michail Tsagris and Konstantinos Fokianos.

References

Armillotta, M. and K. Fokianos (2022a). Poisson network autoregression. https://arxiv.org/abs/2104.06296

Armillotta, M. and K. Fokianos (2022b). Testing linearity for network autoregressive models. https://arxiv.org/abs/2202.03852

See Also

`global_optimise_LM_tnarpq`,`global_optimise_LM_stnarpq`, `score_test_stnarpq_j`

Examples

```
# load data
data(crime)
data(crime_W)
#estimate linear PNAR model
```

```
mod1 <- lin_estimnarpq(crime, crime_W, p = 2)
b <- mod1$coeflin[, 1]

g <- global_optimise_LM_tnarpq(b = b, y = crime, W = crime_W, p = 2, d = 1)
supg <- g$supLM
score_test_tnarpq_j(supLM = supg, b = b, y = crime, W = crime_W, p = 2, d = 1, J = 5)
```

Index