Package ‘spGARCH’

May 9, 2026

Type Package

Title Spatial ARCH and GARCH Models (spGARCH)

Version 0.2.3

Maintainer Philipp Otto <`philipp.otto89@gmail.com`>

Contact <`philipp.otto89@gmail.com`>

Description

A collection of functions to deal with spatial and spatiotemporal autoregressive conditional heteroscedasticity (spatial ARCH and GARCH models) by Otto, Schmid, Garthoff (2018, Spatial Statistics) <doi:10.1016/j.spasta.2018.07.005>: simulation of spatial ARCH-type processes (spARCH, log/exponential-spARCH, complex-spARCH); quasi-maximum-likelihood estimation of the parameters of spARCH models and spatial autoregressive models with spARCH disturbances, diagnostic checks, visualizations.

License GPL

Encoding UTF-8

Depends R (>= 3.1.0)

Imports Rcpp (>= 0.12.4), stats, truncnorm, Rsolnp, spdep, Matrix, nleqslv, methods, crayon

LinkingTo Rcpp, RcppEigen, Matrix

NeedsCompilation yes

Author Philipp Otto [cre, aut] (ORCID: <`https://orcid.org/0000-0002-9796-6682`>)

Repository CRAN

Date/Publication 2025-04-09 11:40:01 UTC

Contents

`extractAIC.spARCH` *Computes the Akaike information criterion*

Description

The function extracts the log-likelihood of a spatial ARCH model.

Usage

```
## S3 method for class 'spARCH'
extractAIC(fit, scale, k = 2, ...)
```

Arguments

`fit`	`spARCH` object (fitted model resulting of `qml.spARCH`).
`scale`	currently unused for `spARCH` objects.
`k`	parameter specifying the weight for the penalizing term.
`...`	Other arguments.

Details

Numeric vector of length 2 is returned. The first element specify the `edf` (equivalent degree of freedom) and the Akaike information criterion is returned as second element.

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

`fitted.spARCH` *Extract model fitted values*

Description

The function extracts the fitted values of a spatial ARCH model.

Usage

```
## S3 method for class 'spARCH'
## S3 method for class 'spARCH'
fitted(object, ...)
```

Arguments

`object`	`spARCH` object generated by `qml.spARCH` or `qml.SARspARCH`.
`...`	Other arguments.

Details

Fitted values extracted from the `object`.

Author(s)

Philipp Otto `<philipp.otto@glasgow.ac.uk>`

See Also

`residuals.spARCH`, `logLik.spARCH`.

`logLik.spARCH` *Extract logarithmic likelihood*

Description

The function extracts the log-likelihood of a spatial ARCH model.

Usage

```
## S3 method for class 'spARCH'
## S3 method for class 'spARCH'
logLik(object, ...)
```

Arguments

`object`	`spARCH` Object of `qml.spARCH`.
`...`	Other arguments.

Details

`logLik` object is returned.

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

See Also

`residuals.spARCH`, `fitted.spARCH`.

`plot.spARCH` *Descriptive plots for residuals of a fitted spatial ARCH model*

Description

The function depicts several descriptive statistics of the residuals of a fitted spatial ARCH model.

Usage

```
## S3 method for class 'spARCH'
## S3 method for class 'spARCH'
plot(x, which = c(1:3), ask, ..., qqline = TRUE)
```

Arguments

`x`	spARCH object generated by `qml.spARCH` or `qml.SARspARCH`.
`which`	Index number of plot to be returned.
`ask`	if `TRUE`, the user is asked before plotting the next figure.
`...`	Other arguments.
`qqline`	A line of the normal distribution for comparison is added to the Q-Q plot, if `qqline = TRUE`.

Details

The function `plot.spARCH` provides several descriptive plots to analyze the residuals of a fitted spatial ARCH model, namely (1) Moran's plot for residuals, (2) Moran's plot for squared residuals, and (3) Normal Q-Q plot for standardized residuals.

Note

For details of `moran.plot` see: `help(moran.plot)`.

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

`prostate_cancer` *Logarithmic incidence rates of prostate cancer and covariates*

Description

The dataset contains logarithmic incidence rates from the National Cancer Institute and Centers for Disease, Control and Prevention (State Cancer Profiles) and factor loadings of a set of covariates. The incidence rates are 5-year averages from 2008 to 2012 in several southeastern states (Arkansas, Louisiana, Mississippi, Tennessee, North and South Carolina, Georgia, Alabama, and Florida). Missing values were imputed by spatial averaging.

Usage

```
data("prostate_cancer")
```

Format

A list with three entries:

`data`	a data frame; see below for details
`B`	a numeric matrix; weighting matrix B to run the example
`W`	a numeric matrix; weighting matrix W to run the example

The data frame contains 755 observations of the following 12 variables.

`log_incidence_rates`	a numeric vector; logarithmic incidence rates of prostate cancer
`F_1`	a numeric vector; scores of factor 1 (environment: fine atmospheric particles and aerosols)
`F_2`	a numeric vector; scores of factor 2 (environment: particulate matter)
`F_3`	a numeric vector; scores of factor 3 (weather: solar radiation and temperature)
`F_4`	a numeric vector; scores of factor 4 (weather: temperature differences)
`F_5`	a numeric vector; scores of factor 5 (behavior: smoking)
`F_6`	a numeric vector; scores of factor 6 (behavior: drinking)
`F_7`	a numeric vector; scores of factor 7 (behavior: preventive health care)
`F_8`	a numeric vector; scores of factor 8 (behavior: physical activity)
`F_9`	a numeric vector; scores of factor 9 (health: overweight)
`F_10`	a numeric vector; scores of factor 10 (health: cholesterol and blood pressure)
`PSA_test`	a numeric vector; percentage of positive results for a prostate-specific antigen (PSA) test

Source

https://statecancerprofiles.cancer.gov/index.html

National Cancer Institute, Centers for Disease, Control and Prevention

References

https://statecancerprofiles.cancer.gov/map/map.withimage.php?99&001&001&00&0&02&0&1&10
Otto, P. (2019). spGARCH: An R-Package for Spatial and Spatiotemporal ARCH and GARCH models To appear: The R Journal URL: https://arxiv.org/abs/1812.01871

Examples

```
data(prostate_cancer)

## Not run:

# Estimation (long running example)
formula <- "log_incidence_rates ~ F_2 + F_10"
out     <- qml.SARspARCH(formula, B = prostate_cancer$B, W = prostate_cancer$W,
                         type = "spARCH", data = prostate_cancer$data)

# Summary
summary(out)


## End(Not run)
```

`qml.SARspARCH` *Maximum-likelihood estimation of a spatial autoregressive model with spatial ARCH residuals*

Description

The function fits a spatial autoregressive model with spatial ARCH residuals using the maximum-likelihood approach. All parameters are jointly estimated. In addition, external regressor may be included in the mean equation.

Usage

```
qml.SARspARCH(formula, B, W, type = "spARCH", data = NULL,
                b = 2, start = NULL, eigen_v = NULL, control = list())
```

Arguments

`formula`	an object of class "`formula`"): a symbolic description of the model to be fitted. The details of model specification are given under "Details." y must be a numeric vector of length `n` (dimension of the weighting matrix)
`B`	`n` times `n` spatial weight matrix for the spatial autoregressive part
`W`	`n` times `n` spatial weight matrix for the spatial ARCH residuals
`type`	type of spatial ARCH model to be fitted for the error process (see Details)

`data`	an optional data frame, list or environment containing the variables in the model. If not found in data, the variables are taken from the working space.
`b`	parameter `b` for the E-spARCH model (`type = "exp"`), must be a (positive) natural number
`start`	vector of starting values for the numerical optimization of the log-likelihood (optional)
`eigen_v`	eigen values of B (optional)
`control`	list of control variables for iterative maximization of the log-likelihood

Details

For `type = "spARCH"`, the functions fits a simple spatial autoregressive model with spatial ARCH residuals, i.e.,

$$\boldsymbol{Y} = \lambda \mathbf{B} \boldsymbol{Y} + \mathbf{X} \boldsymbol{\beta} + \boldsymbol{h}^{1/2} \boldsymbol{\varepsilon}$$

with

$$\boldsymbol{h} = \alpha \mathbf{1}_n + \rho \mathbf{W} \boldsymbol{Y}^{(2)} \, .$$

The distribution of the error term is assumed to be Gaussian.

If `type = "log-ARCH"`, a spatial log-ARCH process is estimated for the error term, i.e.,

$$\ln(\boldsymbol{h}) = \alpha \mathbf{1}_n + \rho \mathbf{W} g_b(\boldsymbol{\varepsilon}) \, .$$

The function g_b is defined as

$$g_b(\boldsymbol{\varepsilon}) = (\ln |\varepsilon(\boldsymbol{s}_1)|^b, \ldots, \ln |\varepsilon(\boldsymbol{s}_n)|^b)'$$

and the error term is also assumed to be Gaussian.

The modelling equation gan be specified as for `lm`, i.e., as `formula` object. A typical model has the form `response ~ terms` where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for `response`. A terms specification of the form `first + second` indicates all the terms in first together with all the terms in second with duplicates removed. A specification of the form `first:second` indicates the set of terms obtained by taking the interactions of all terms in first with all terms in second. The specification `first*second` indicates the cross of first and second. This is the same as `first + second + first:second`. However, there is no `offset` permitted for the `qml.SARspARCH`.

For an intercept-only model, the `formula` can be specified as `response ~ 1`. In addition, it is possible to fit an intercept-free model with `response ~ 0` or `response ~ 0 + terms`.

To summarize the results of the model fit, use the generic function `summary`. For analysis of the residuals, the generic `plot.spARCH` function provides several descriptive plots. For numerical maximization of the log-likelihood, the function uses the algorithm of `solnp` from the package `Rsolnp`.

Value

A spARCH object with the following elements is returned:

`coefficients`	Parameter estimates α and ρ.
`residuals`	Vector of residuals.

`fitted.values`	Fitted values.
`stderr`	Standard errors of the estimates (Cramer-Rao estimates).
`hessian`	Hessian matrix of the negative Log-Likelihood at the estimated minimum.
`LL`	Value of the Log-Likelihood at the estimated maximum.
`h`	Fitted vector $\boldsymbol{h}$.
`y`	Vector of observations (input values).
`h`	Chosen type (input).
`B`	Spatial weight matrix (input).
`W`	Spatial weight matrix (input).
`regressors`	Are regressors included? `TRUE/FALSE`
`AR`	Is an autoregressive term in the mean equation? `TRUE/FALSE`
`X`	Matrix of regressors if `regressor = TRUE`

Control Arguments

see also: `solnp`

- `rho` - This is used as a penalty weighting scalar for infeasibility in the augmented objective function. The higher its value the more the weighting to bring the solution into the feasible region (default 1). However, very high values might lead to numerical ill conditioning or significantly slow down convergence.
- `outer.iter` - Maximum number of major (outer) iterations (default 400).
- `inner.iter` - Maximum number of minor (inner) iterations (default 800).
- `delta` - Relative step size in forward difference evaluation (default 1.0e-7).
- `tol` - Relative tolerance on feasibility and optimality (default 1e-8).
- `trace` - The value of the objective function and the parameters is printed at every major iteration (default 1).

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

References

Philipp Otto, Wolfgang Schmid, Robert Garthoff (2018). Generalised Spatial and Spatiotemporal Autoregressive Conditional Heteroscedasticity. Spatial Statistics 26, pp. 125-145. doi:10.1016/j.spasta.2018.07.005, arXiv: doi:10.48550/arXiv.1609.00711

See Also

`solnp` (package `Rsolnp`)

Examples

```
require("spdep")
rho <- 0.5
alpha <- 1
lambda <- 0.5
d <- 5
n <- d^2
nblist <- cell2nb(d, d, type = "queen")
W <- nb2mat(nblist)
B <- W

X <- cbind(rep(1, n), rnorm(n))
beta <- c(5, 2)

y <- solve(diag(n) - lambda * B) %*%
   (sim.spARCH(n = n, rho = rho, alpha = alpha, W = W, type = "log-spARCH") + X %*% beta)
y <- as.vector(y)
out <- qml.SARspARCH(y ~ X[,2], B = B, W = W, type = "log-spARCH")

summary(out)
```

`qml.spARCH` *Maximum-likelihood estimation of a spatial ARCH model*

Description

The function fits a spatial ARCH model using the maximum-likelihood approach. In addition, external regressor may be included in the mean equation.

Usage

```
qml.spARCH(formula, W, type = "spARCH", data = NULL,
             b = 2, start = NULL, control = list())
```

Arguments

`formula`	an object of class "`formula`"): a symbolic description of the model to be fitted. The details of model specification are given under "Details."" `y` must be a numeric vector of length `n` (dimension of the weighting matrix)
`W`	`n` times `n` spatial weight matrix
`type`	type of spatial ARCH model to be fitted (see Details)
`data`	an optional data frame, list or environment containing the variables in the model. If not found in data, the variables are taken from the working space.
`b`	parameter `b` for the E-spARCH model (`type = "exp"`), must be a (positive) natural number
`start`	vector of starting values for the numerical optimization of the log-likelihood (optional)
`control`	list of control variables for iterative maximization of the log-likelihood

Details

For `type = "spARCH"`, the functions fits a simple spatial ARCH model with one spatial lag, i.e.,

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{h}^{1/2}\boldsymbol{\varepsilon}$$

with

$$\boldsymbol{h} = \alpha \mathbf{1}_n + \rho \mathbf{W} \boldsymbol{Y}^{(2)} \,.$$

The distribution of the error term is assumed to be Gaussian.

If `type = "log-spARCH"`, a spatial log-ARCH process is estimated, i.e.,

$$\ln(\boldsymbol{h}) = \alpha \mathbf{1}_n + \rho \mathbf{W} g_b(\boldsymbol{\varepsilon}) \,.$$

The function g_b is defined as

$$g_b(\boldsymbol{\varepsilon}) = (\ln |\varepsilon(\boldsymbol{s}_1)|^b, \ldots, \ln |\varepsilon(\boldsymbol{s}_n)|^b)'$$

and the error term is also assumed to be Gaussian.

The modelling equation gan be specified as for `lm`, i.e., as `formula` object. A typical model has the form `response ~ terms` where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for `response`. A terms specification of the form `first + second` indicates all the terms in first together with all the terms in second with duplicates removed. A specification of the form `first:second` indicates the set of terms obtained by taking the interactions of all terms in first with all terms in second. The specification `first*second` indicates the cross of first and second. This is the same as `first + second + first:second`. However, there is no `offset` permitted for the `qml.spARCH`.

For an intercept-only model, the `formula` can be specified as `response ~ 1`. In addition, it is possible to fit an intercept-free model with `response ~ 0` or `response ~ 0 + terms`.

To summarize the results of the model fit, use the generic function `summary`. For analysis of the residuals, the generic `plot` provides several descriptive plots. For numerical maximization of the log-likelihood, the function uses the algorithm of `solnp` from the package `Rsolnp`.

Value

A spARCH object with the following elements is returned:

`coefficients`	Parameter estimates α and ρ.
`residuals`	Vector of residuals.
`fitted.values`	Fitted values.
`stderr`	Standard errors of the estimates (Cramer-Rao estimates).
`hessian`	Hessian matrix of the negative Log-Likelihood at the estimated minimum.
`LL`	Value of the Log-Likelihood at the estimated maximum.
`h`	Fitted vector $\boldsymbol{h}$.
`y`	Vector of observations (input values).
`h`	Chosen type (input).
`W`	Spatial weight matrix (input).
`regressors`	Are regressors included? `TRUE/FALSE`
`AR`	Is an autoregressive term in the mean equation? `TRUE/FALSE`
`X`	Matrix of regressors if `regressor = TRUE`

Control Arguments

see also: `solnp`

- `rho` - This is used as a penalty weighting scalar for infeasibility in the augmented objective function. The higher its value the more the weighting to bring the solution into the feasible region (default 1). However, very high values might lead to numerical ill conditioning or significantly slow down convergence.
- `outer.iter` - Maximum number of major (outer) iterations (default 400).
- `inner.iter` - Maximum number of minor (inner) iterations (default 800).
- `delta` - Relative step size in forward difference evaluation (default 1.0e-7).
- `tol` - Relative tolerance on feasibility and optimality (default 1e-8).
- `trace` - The value of the objective function and the parameters is printed at every major iteration (default 1).

Author(s)

Philipp Otto `<philipp.otto@glasgow.ac.uk>`

References

Philipp Otto, Wolfgang Schmid, Robert Garthoff (2018). Generalised Spatial and Spatiotemporal Autoregressive Conditional Heteroscedasticity. Spatial Statistics 26, pp. 125-145. doi:10.1016/j.spasta.2018.07.005, arXiv: doi:10.48550/arXiv.1609.00711

See Also

`solnp` (package `Rsolnp`)

Examples

```
require("spdep")

# directional spatial ARCH process (W is triangular, 1:1 origin)

rho <- 0.5
alpha <- 1
d <- 5
n <- d^2
nblist <- cell2nb(d, d, type = "queen")
W <- nb2mat(nblist)
W[lower.tri(W)] <- 0
y <- sim.spARCH(n = n, rho = rho, alpha = alpha, W = W, type = "spARCH")

out <- qml.spARCH(y ~ 0, W = W, type = "spARCH")

summary(out)
```

`residuals.spARCH` *Extract model residuals*

Description

The function extracts the residuals of a fitted spatial ARCH model.

Usage

```
## S3 method for class 'spARCH'
## S3 method for class 'spARCH'
residuals(object, ...)
```

Arguments

`object`	`spARCH` object generated by `qml.spARCH` or `qml.SARspARCH`.
`...`	Other arguments.

Details

The function extracts the residuals of a fitted spatial ARCH model.

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

See Also

`residuals.spARCH`, `fitted.spARCH`.

`sim.spARCH` *Simulation of spatial ARCH models*

Description

The function generates `n` random numbers of a spatial ARCH process for given parameters and weighting schemes.

Usage

```
sim.spARCH(n = dim(W)[1], rho, alpha, W, b = 2, type = "spARCH", control = list())
```

Arguments

`n`	number of observations. If `length(n)` > 1, the length is taken to be the number required. Default `dim(W)[1]`
`rho`	spatial dependence parameter rho
`alpha`	unconditional variance level alpha
`W`	`n` times `n` spatial weight matrix
`b`	parameter `b` for logarithmic spatial ARCH (only needed if `type = "log-spARCH"`). Default 2.
`type`	type of simulated spARCH process (see details)
`control`	list of control arguments (see below)

Details

The function simulates `n` observations $Y = (Y_1, ..., Y_n)'$ of a spatial ARCH process, i.e.,

$$\boldsymbol{Y} = diag(\boldsymbol{h})^{1/2}\boldsymbol{\varepsilon}\,,$$

where $\boldsymbol{\varepsilon}$ is a spatial White Noise process. The definition of $\boldsymbol{h}$ depends on the chosen `type`. The following types are available.

- `type = "spARCH"` - simulates $\boldsymbol{\varepsilon}$ from a truncated normal distribution on the interval $[-a, a]$, such that $\boldsymbol{h} > 0$ with

  $$\boldsymbol{h} = \alpha + \rho\mathbf{W}\boldsymbol{Y}^{(2)} \text{ and } a = 1/||\rho^2\mathbf{W}^2||_1^{1/4}.$$

  Note that the normal distribution is not trunctated ($a = \infty$), if $\mathbf{W}$ is a strictly triangular matrix, as it is ensured that $\boldsymbol{h} > \mathbf{0}$. Generally, it is sufficient that if there exists a permutation such that $\mathbf{W}$ is strictly triangular. In this case, the process is called oriented spARCH process.

- `type = "log-spARCH"` - simulates a logarithmic spARCH process (log-spARCH), i.e.,

  $$\ln \boldsymbol{h} = \alpha + \rho\mathbf{W}g(\boldsymbol{\varepsilon})\,.$$

  For the log-spARCH process, the errors follow a standard normal distribution. The function g_b is given by

  $$g_b(\boldsymbol{\varepsilon}) = (\ln|\varepsilon(\boldsymbol{s}_1)|^b, \ldots, \ln|\varepsilon(\boldsymbol{s}_n)|^b)'\,.$$

- `type = "complex-spARCH"` - allows for complex solutions of $\boldsymbol{h}^{1/2}$ with

  $$\boldsymbol{h} = \alpha + \rho\mathbf{W}\boldsymbol{Y}^{(2)}\,.$$

  The errors follow a standard normal distribution.

Value

The functions returns a vector $\boldsymbol{y}$.

Control Arguments

- `seed` - positive integer to initialize the random number generator (RNG), default value is a random integer in $[1, 10^6]$
- `silent` - if `FALSE`, current random seed is reported
- `triangular` - if `TRUE`, $\mathbf{W}$ is a triangular matrix and there are no checks to verify this assumption (default `FALSE`)

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

References

Philipp Otto, Wolfgang Schmid, Robert Garthoff (2018). Generalised Spatial and Spatiotemporal Autoregressive Conditional Heteroscedasticity. Spatial Statistics 26, pp. 125-145. doi:10.1016/j.spasta.2018.07.005, arXiv: doi:10.48550/arXiv.1609.00711

Examples

```
require("spdep")

# 1st example
##############

# parameters

rho <- 0.5
alpha <- 1
d <- 2

nblist <- cell2nb(d, d, type = "queen")
W <- nb2mat(nblist)

# simulation

Y <- sim.spARCH(rho = rho, alpha = alpha, W = W, type = "log-spARCH")

# visualization

image(1:d, 1:d, array(Y, dim = c(d,d)), xlab = expression(s[1]), ylab = expression(s[2]))

# 2nd example
##############

# two spatial weighting matrices W_1 and W_2
# h = alpha + rho_1 W_1 Y^2 + rho_2 W_2 Y^2

W_1 <- W
nblist <- cell2nb(d, d, type = "rook")
W_2 <- nb2mat(nblist)
```

```
rho_1 <- 0.3
rho_2 <- 0.7

W <- rho_1 * W_1 + rho_2 * W_2
rho <- 1

Y <- sim.spARCH(n = d^2, rho = rho, alpha = alpha, W = W, type = "log-spARCH")
image(1:d, 1:d, array(Y, dim = c(d,d)), xlab = expression(s[1]), ylab = expression(s[2]))
```

`sim.spGARCH` *Simulation of spatial ARCH models*

Description

The function generates `n` random numbers of a spatial GARCH process for given parameters and weighting schemes.

Usage

```
sim.spGARCH(n = dim(W1)[1], rho, lambda, alpha, W1, W2,
            b = 2, zeta = 0.5, theta = 0.5, type = "spGARCH",
            control = list())
```

Arguments

`n`	number of observations. If `length(n)` > 1, the length is taken to be the number required. Default `dim(W1)[1]`
`rho`	spatial dependence parameter rho
`lambda`	spatial dependence parameter lambda
`alpha`	unconditional variance level alpha
`W1`	`n` times `n` spatial weight matrix (ARCH component, parameter rho)
`W2`	`n` times `n` spatial weight matrix (GARCH component, parameter lambda)
`b`	parameter `b` for logarithmic spatial GARCH (only needed if `type = "log-spGARCH"`). Default 2.
`zeta`	parameter `zeta` for exponential spatial GARCH (only needed if `type = "e-spGARCH"`). Default 0.5.
`theta`	parameter `theta` for exponential spatial GARCH (only needed if `type = "e-spGARCH"`). Default 0.5.
`type`	type of simulated spGARCH process (see details)
`control`	list of control arguments (see below)

Details

The function simulates `n` observations $Y = (Y_1, ..., Y_n)'$ of a spatial GARCH process, i.e.,

$$\boldsymbol{Y} = diag(\boldsymbol{h})^{1/2}\boldsymbol{\varepsilon}\,,$$

where $\boldsymbol{\varepsilon}$ is a spatial White Noise process. The definition of $\boldsymbol{h}$ depends on the chosen `type`. The following types are available.

- `type = "spGARCH"` - simulates $\boldsymbol{\varepsilon}$ from a truncated normal distribution on the interval $[-a, a]$, such that $\boldsymbol{h} > 0$ with

  $$\boldsymbol{h} = \alpha + \rho \mathbf{W}_1 \boldsymbol{Y}^{(2)} + \lambda \mathbf{W}_2 \boldsymbol{h} \text{ and } a = 1/||\rho^2 \mathbf{W}_1^2||_1^{1/4}.$$

  Note that the normal distribution is not trunctated ($a = \infty$), if $\mathbf{W}_1$ is a strictly triangular matrix, as it is ensured that $\boldsymbol{h} > \mathbf{0}$. Generally, it is sufficient that if there exists a permutation such that $\mathbf{W}_1$ is strictly triangular. In this case, the process is called oriented spGARCH process.
- `type = "e-spGARCH"` - simulates an exponential spARCH process (e-spGARCH), i.e.,

  $$\ln \boldsymbol{h} = \alpha + \rho \mathbf{W}_1 g(\boldsymbol{\varepsilon}) + \lambda \mathbf{W}_2 log(\boldsymbol{h})\,.$$

  For the e-spGARCH process, the errors follow a standard normal distribution. The function g is given by

  $$g(\boldsymbol{\varepsilon}) = \Theta \varepsilon + \zeta(|\varepsilon| - E(|\varepsilon|))\,.$$
- `type = "log-spGARCH"` - simulates a logarithmic spARCH process (log-spGARCH), i.e.,

  $$\ln \boldsymbol{h} = \alpha + \rho \mathbf{W}_1 g(\boldsymbol{\varepsilon}) + \lambda \mathbf{W}_2 log(\boldsymbol{h})\,.$$

  For the log-spGARCH process, the errors follow a standard normal distribution. The function g is given by

  $$g(\boldsymbol{\varepsilon}) = (\ln|\varepsilon(\boldsymbol{s}_1)|^b, \ldots, \ln|\varepsilon(\boldsymbol{s}_n)|^b)'\,.$$
- `type = "complex-spGARCH"` - allows for complex solutions of $\boldsymbol{h}^{1/2}$ with

  $$\boldsymbol{h} = \alpha + \rho \mathbf{W}_1 \boldsymbol{Y}^{(2)} + \lambda \mathbf{W}_2 \boldsymbol{h}\,.$$

  The errors follow a standard normal distribution.

Value

The functions returns a vector $\boldsymbol{y}$.

Control Arguments

- `seed` - positive integer to initialize the random number generator (RNG), default value is a random integer in $[1, 10^6]$
- `silent` - if `FALSE`, current random seed is reported
- `triangular` - if `TRUE`, $\mathbf{W}$ is a triangular matrix and there are no checks to verify this assumption (default `FALSE`)

Author(s)

Philipp Otto `<philipp.otto@glasgow.ac.uk>`

References

Philipp Otto, Wolfgang Schmid (2019). Spatial GARCH Models - A Unified Approach. arXiv: doi:10.48550/arXiv.1908.08320

Examples

```
require("spdep")

# 1st example (spatial GARCH)
##############

# parameters

rho    <- 0.5
lambda <- 0.3
alpha  <- 1
d      <- 5

nblist <- cell2nb(d, d, type = "rook") # lattice process with Rook's contiguity matrix
W_1    <- nb2mat(nblist)
W_2    <- W_1

# simulation

Y      <- sim.spGARCH(rho = rho, lambda = lambda, alpha = alpha,
                      W1 = W_1, W2 = W_2, type = "spGARCH")

# visualization

image(1:d, 1:d, array(Y, dim = c(d,d)), xlab = expression(s[1]), ylab = expression(s[2]))

# 2nd example (exponential spatial GARCH)
##############

# parameters

rho    <- 0.5
lambda <- 0.3
alpha  <- 1
zeta   <- 0.5
theta  <- 0.5
d      <- 5

nblist <- cell2nb(d, d, type = "rook") # lattice process with Rook's contiguity matrix
W_1    <- nb2mat(nblist)
W_2    <- W_1

# simulation
```

```
Y       <- sim.spGARCH(rho = rho, lambda = lambda, alpha = alpha,
                      W1 = W_1, W2 = W_2, zeta = zeta, theta = 0.5, type = "e-spGARCH")

# visualization

image(1:d, 1:d, array(Y, dim = c(d,d)), xlab = expression(s[1]), ylab = expression(s[2]))
```

`spARCH` *Output of Quasi-Maximum-Likelihood Estimation*

Description

The `spARCH` class is a class generated by the estimation functions `qml.spARCH` or `qml.SARspARCH` comprising the results of the quasi-maximum-likelihood estimation.

Methods

Several generic methods for summarizing the results are available:

`plot.spARCH` - Descriptive plots for residuals of a fitted spatial ARCH model

`plot(spARCH_object)``summary(spARCH_object)` `summary.spARCH` - The function returns a summary of the model fit of a spatial ARCH model

`fitted(spARCH_object)` `fitted.spARCH` - The function extracts the fitted values of a spatial ARCH model.

`spGARCH` *spGARCH - Package to fit spatial ARCH models*

Description

A collection of functions for simulating and fitting spatial autoregressive conditional heteroscedasticity (spARCH) processes are provided.

The functions `sim.spARCH` and `sim.spGARCH` are the main function for simulating spARCH and spGARCH processes, respectively. Via the argument `type` several types of spatial ARCH and GARCH can be simulated, e.g., exponential spARCH models, spARCH for oriented processes, or spARCH processes with truncated error support. For details, refer the paper Otto, Schmid, and Garthoff (2018), see doi:10.1016/j.spasta.2018.07.005. Moreover, the package provides function for fitting spARCH models. Basically, there are two functions to fit these kind of model: `qml.spARCH` and `qml.SARspARCH`. First, spARCH models can be fitted by `qml.spARCH`.

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

References

Philipp Otto, Wolfgang Schmid, Robert Garthoff (2018). Generalised Spatial and Spatiotemporal Autoregressive Conditional Heteroscedasticity. Spatial Statistics 26, pp. 125-145. doi:10.1016/j.spasta.2018.07.005, arXiv: doi:10.48550/arXiv.1609.00711

`summary.spARCH` *Summary for spARCH object*

Description

The function returns a summary of the model fit of a spatial ARCH model (`qml.spARCH` or `qml.SARspARCH`).

Usage

```
## S3 method for class 'spARCH'
summary(object, ...)
## S3 method for class 'summary.spARCH'
print(x, digits = max(5, .Options$digits - 3),
signif.stars = TRUE, ...)
## S3 method for class 'spARCH'
print(x, ...)
```

Arguments

`object`	spARCH object generated by `qml.spARCH` or `qml.SARspARCH`.
`digits`	The number of significant digits to be printed.
`signif.stars`	Logical variable. If TRUE, significance stars are printed for each coefficient.
`x`	spARCH object of `qml.spARCH`.
`...`	further arguments passed to or from other methods

Details

The function `summary.spARCH` returns an spARCH object with all results (coefficients, residuals, diagnostic checks etc.). If the returned object is printed, a detailed summary of the model fit is returned.

Value

The function returns the input spARCH object, plus

`Coef`	a matrix with columns for the estimated coefficients, their standard error, t-statistics and corresponding (two-sided, asymptotic) p-values.
`AIC`	Akaike information criterion
`BIC`	Bayesian Schwarz information criterion
`moran_res`	Test on spatial spatial autocorrelation of the residuals (based on Morans I, `moran.test`).
`moran_sq_res`	Test on spatial spatial autocorrelation of the squared residuals (based on Morans I, `moran.test`).

Note

For further details about the Moran's I test see `moran.test`.

Author(s)

Philipp Otto <`philipp.otto@glasgow.ac.uk`>

See Also

The model fitting functions `qml.spARCH` and `qml.SARspARCH`. Function `coef` will extract the matrix of coefficients with standard errors, t-statistics and p-values.

Examples

```
require("spdep")

# directional spatial ARCH process (W is triangular, 1:1 origin)

rho <- 0.5
alpha <- 1
d <- 5
n <- d^2
nblist <- cell2nb(d, d, type = "queen")
W <- nb2mat(nblist)
W[lower.tri(W)] <- 0
y <- sim.spARCH(n = n, rho = rho, alpha = alpha, W = W, type = "spARCH")

out <- qml.spARCH(y ~ 0, W = W, type = "spARCH")

summary(out)
```

Index