

Package ‘uGMAR’

September 22, 2018

Title Estimate Univariate Gaussian or Student's t Mixture
Autoregressive Model

Version 3.0.1

Description Maximum likelihood estimation of univariate Gaussian Mixture Autoregressive (GMAR), Student's t Mixture Autoregressive (StMAR) and Gaussian and Student's t Mixture Autoregressive (G-StMAR) models, quantile residual tests, graphical diagnostics, forecast and simulate from GMAR, StMAR and G-StMAR processes.

Also general linear constraints and restricting autoregressive parameters to be the same for all regimes are supported.

Leena Kalliovirta, Mika Meitz, Pentti Saikkonen (2015) <doi:10.1111/jtsa.12108>,

Mika Meitz, Daniel Preve, Pentti Saikkonen (2018) <arXiv:1805.04010>.

Depends R (>= 3.4.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports Brodtingnag (>= 1.2-4), parallel, pbapply (>= 1.3-2), stats (>= 3.3.2)

Suggests gsl (>= 1.9-10.3), testthat, knitr, rmarkdown

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Savi Virolainen [aut, cre]

Maintainer Savi Virolainen <savi.virolainen@helsinki.fi>

Repository CRAN

Date/Publication 2018-09-22 12:00:02 UTC

R topics documented:

aa_uGMAR	3
add_data	3

add_dfs	4
all_pos_ints	5
calc_gradient	6
changeRegime	7
change_parametrization	9
checkAndCorrectData	11
checkConstraintMat	11
checkPM	12
check_data	13
check_gsmar	13
check_model	14
check_params_length	14
diagnosticPlot	15
extractRegime	18
fitGMAR	20
fitGSMAR	20
forecastGMAR	24
format_valuef	25
GAfit	25
getOmega	28
get_ar_roots	30
get_IC	31
get_regime_means	31
GSMAR	32
isStationary	36
isStationary_int	38
iterate_more	40
loglikelihood	41
loglikelihood_int	44
mixingWeights	47
mixingWeights_int	49
nParams	51
parameterChecks	52
pick_alphas	53
pick_dfs	55
pick_pars	56
pick_phi0	57
plot.gsmarpred	59
plot.qrtest	59
plotGMAR	62
predict.gsmar	62
print.gsmarpred	64
print.gsmarsum	65
quantileResiduals	65
quantileResiduals_int	68
randomIndividual	70
randomIndividual_int	73
random_regime	75

reformConstrainedPars	76
reformParameters	78
reformRestrictedPars	79
regime_distance	80
removeAllConstraints	81
simulateGMAR	82
simulateGSMAR	83
sortComponents	85
standardErrors	86
swap_parametrization	88
VIX	89

Index	90
--------------	-----------

aa_uGMAR	<i>uGMAR: Estimate Univariate Gaussian or Student's t Mixture Autoregressive Model</i>
----------	--

Description

uGMAR is a package for estimating univariate Gaussian Mixture Autoregressive (GMAR), Student's t Mixture Autoregressive (StMAR) and Gaussian and Student's t Mixture Autoregressive (G-StMAR) models. It provides functions for quantile residuals tests, graphical diagnostics, forecasting and simulations. Applying general linear constraints to the autoregressive parameters, or restricting them to be the same for all regimes is supported.

Many of the functions documented are not exported, but for internal use only. The vignette is a good place to start.

add_data	<i>Add data to object of class 'gsmar' defining a GMAR, StMAR or G-StMAR model</i>
----------	--

Description

add_data adds or updates data to object of class 'gsmar' that defines a GMAR, StMAR or G-StMAR model based on the given object. Also calculates mixing weights and quantile residuals accordingly.

Usage

```
add_data(data, gsmar, calc_qresiduals = TRUE, calc_std_errors = FALSE)
```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
gsmar	object of class 'gsmar', generated by function fitGSMAR() or GSMAR().
calc_qresiduals	should quantile residuals be calculated? Default is TRUE if the model contains data.
calc_std_errors	should approximate standard errors be calculated?

Value

Returns an object of class 'gsmar' defining the GMAR, StMAR or G-StMAR model with the data added to the model. If the object already contained data, the data will be updated.

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

[fitGSMAR](#), [GSMAR](#), [iterate_more](#), [get_gradient](#), [get_regime_means](#)

Examples

```
# GMAR model without data
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(p=1, M=3, params=params13, model="GMAR")
gmar13

# Add data to the model
gmar13 <- add_data(data=VIX, gmar13)
gmar13
```

add_dfs

Add random dfs to a vector

Description

add_dfs adds random dfs to a vector

Usage

```
add_dfs(x, how_many)
```

Arguments

x a vector to add the dfs to
how_many how many dfs?

Details

For details read the source code.

Value

Returns `c(ind, dfs)` with `how_many` dfs-elements.

`all_pos_ints` *Check whether all arguments are positive scalar whole numbers*

Description

`all_pos_ints` tells whether all the elements in a vector are strictly positive whole numbers.

Usage

```
all_pos_ints(x)
```

Arguments

x a vector containing the elements to be tested.

Value

Returns TRUE or FALSE accordingly.

`calc_gradient`*Calculate gradient or Hessian matrix*

Description

`calc_gradient` or `calc_hessian` calculates the gradient or Hessian matrix of the given function at the given point using central difference numerical approximation. `get_gradient` or `get_hessian` calculates the gradient or Hessian matrix of the log-likelihood function at the parameter estimates of class 'gsmar' object.

Usage

```
calc_gradient(x, fn, h = 6e-06, ...)
```

```
calc_hessian(x, fn, h = 6e-06, ...)
```

```
get_gradient(gsmar, h = 6e-06)
```

```
get_hessian(gsmar, h = 6e-06)
```

Arguments

<code>x</code>	a numeric vector specifying the point where the gradient or Hessian should be calculated.
<code>fn</code>	a function that takes in argument <code>x</code> as the first argument.
<code>h</code>	difference used to approximate the derivatives.
<code>...</code>	other arguments passed to <code>fn</code>
<code>gsmar</code>	object of class 'gsmar', generated by function <code>fitGSMAR()</code> or <code>GSMAR()</code> .

Details

Especially the functions `get_gradient()` or `get_hessian()` can be used to check whether the found estimates denote a (local) maximum point, a saddle point or something else.

Value

Gradient functions return numerical approximation of the gradient, and Hessian functions return numerical approximation of the Hessian.

Warning

No argument checks!

Examples

```
# Simple function
foo <- function(x) x^2 + x
calc_gradient(x=1, fn=foo)
calc_gradient(x=-0.5, fn=foo)

# More complicated function
foo <- function(x, a, b) a*x[1]^2 - b*x[2]^2
calc_gradient(x=c(1, 2), fn=foo, a=0.3, b=0.1)

# GMAR model:
params12 <- c(1.12321, 0.90557, 0.29288,
  4.53131, 0.70309, 3.21154, 0.83888)
gmar12 <- GSMAR(VIX, 1, 2, params12)
get_gradient(gmar12)
get_hessian(gmar12)
```

changeRegime	<i>Change the specified regime of parameter vector to the given regime-parameter vector</i>
--------------	---

Description

changeRegime changes the specified regime of the parameter vector to correspond the given regime-parameter vector and returns the modified parameter vector. Does not affect mixing weight parameters.

Usage

```
changeRegime(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, regimeParams, regime)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	<p>For GMAR and StMAR models: a positive integer specifying the number of mixture components.</p> <p>For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p>

	<p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p>
	<p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.</p>
regimeParams	a numeric vector specifying the parameter values that should be inserted to the specified regime. <p>For non-restricted models: For GMAR model: Size $(p+2 \times 1)$ vector $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2)$.</p> <p>For StMAR model: Size $(p+3 \times 1)$ vector $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2, \nu_m)$.</p> <p>For G-StMAR model: Same as GMAR for GMAR-components and same as StMAR for StMAR-components.</p> <p>With linear constraints: Parameter vector as described above, but vector ϕ_m replaced with vector ψ_m that satisfies $\phi_m = R_m \psi_m$.</p> <p>For restricted models: For GMAR model: Size (2×1) vector $(\phi_{m,0}, \sigma_m^2)$.</p> <p>For StMAR model: Size (3×1) vector $(\phi_{m,0}, \sigma_m^2, \nu_m)$.</p> <p>For G-StMAR model: Same as GMAR for GMAR-components and same as StMAR for StMAR-components.</p> <p>With linear constraints: Parameter vector as described above.</p>
regime	a positive integer in the interval [1, M] defining which regime should be changed.

Value

Returns modified parameter vector of the form described in params.

change_parametrization

Change parametrization of the parameter vector

Description

change_parametrization changes the parametrization of the given parameter vector to change_to.

Usage

```
change_parametrization(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, change_to = c("intercept",
  "mean"))
```

Arguments

p a positive integer specifying the order of AR coefficients.

M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

params a real valued parameter vector specifying the model.

For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.
change_to	either "intercept" or "mean" specifying to which parametrization it should be switched to. If set to "intercept", it's assumed that params is mean-parametrized, and if set to "mean" it's assumed that params is intercept-parametrized.

Value

Returns parameter vector described in params, but with parametrization changed from intercept to mean (when change_to=="mean") or from mean to intercept (when change_to=="intercept").

Warning

No argument checks!

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

checkAndCorrectData *Check the data is set correctly and correct if not*

Description

checkAndCorrectData checks that the data is set correctly and corrects it if not. Throws an error if it can't convert the data to the correct form.

Usage

```
checkAndCorrectData(data, p)
```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.

Value

Returns a numeric column matrix containing the data.

checkConstraintMat *Check the constraint matrices*

Description

checkConstraintMat checks for some parts that the constraint matrices are correctly set.

Usage

```
checkConstraintMat(p, M, restricted = FALSE, constraints = NULL)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters.

For non-restricted models: a list of size (pq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: a size (pq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = \psi_1, \dots, \psi_q$.

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

Value

Doesn't return anything, but throws an informative error if finds out that something is wrong.

checkPM	<i>Check p and M are correctly set</i>
---------	--

Description

checkPM checks that the arguments p and M are correctly set.

Usage

```
checkPM(p, M, model = c("GMAR", "StMAR", "G-StMAR"))
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2×1) vector specifying the number of <i>GMAR-type</i> components $M1$ in the first element and <i>StMAR-type</i> components $M2$ in the second. The total number of mixture components is $M = M1 + M2$.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first $M1$ components are <i>GMAR-type</i> and the rest $M2$ components are <i>StMAR-type</i> .

Value

Doesn't return anything, but throws an informative error if something is wrong.

check_data	<i>Check that given object contains data</i>
------------	--

Description

check_data checks that that given object contains data.

Usage

```
check_data(object)
```

Arguments

object an object to be tested

Value

Doesn't return anything, but throws and error if something is wrong.

check_gsmar	<i>Check that given object has class attribute 'gsmar'</i>
-------------	--

Description

check_gsmar checks that that given object has class attribute 'gsmar'.

Usage

```
check_gsmar(object)
```

Arguments

object an object to be tested

Value

Doesn't return anything, but throws and error if something is wrong.

check_model	<i>Check that the argument model is correctly specified.</i>
-------------	--

Description

check_model checks that the argument model is correctly specified.

Usage

```
check_model(model)
```

Arguments

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
-------	--

Value

Doesn't return anything, but throws an error if something is wrong.

check_params_length	<i>Check that the parameter vector has the correct dimension</i>
---------------------	--

Description

check_model checks that the parameter vector has the correct dimension

Usage

```
check_params_length(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
params	a real valued parameter vector specifying the model. For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4)-1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3)+M2-1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices \mathbf{C} that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M+p-1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M+M2+p-1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

Value

Doesn't return anything, but throws an error if something is wrong.

diagnosticPlot

Quantile residual based diagnostic plots for GMAR, StMAR and G-StMAR models

Description

diagnosticPlot plots quantile residual time series, normal QQ-plot, autocorrelation function and squared quantile residual autocorrelation function. There is an option to also plot the individual statistics associated with the quantile residual tests (for autocorrelation and conditional heteroskedasticity) divided by their approximate standard errors with their approximate 95% critical bounds.

Usage

```
diagnosticPlot(gsmar, nlags = 20, nsimu = 2000, plot_indstats = FALSE)
```

Arguments

gsmar	object of class 'gsmar', generated by function fitGSMAR() or GSMAR().
nlags	a positive integer specifying how many lags should be calculated for the autocorrelation and conditional heteroscedasticity statistics.
nsimu	a positive integer specifying to how many simulated values from the process the covariance matrix "Omega" (used to compute the tests) should be based on. Larger number of simulations may result more reliable tests. If smaller than data size, then it will be based on the given data. Ignored if plot_indstats==FALSE.
plot_indstats	set TRUE if the individual statistics discussed in Kalliovirta (2012) should be plotted with their approximate 95% critical bounds (this may take some time).

Details

Sometimes the individual statistics are not plotted because it's not (numerically) possible for to calculate all the necessary estimates required. This may suggest that the model is misspecified.

The dashed lines plotted with autocorrelation functions (for quantile residuals and their squares) are plus-minus $1.96 * T^{-1/2}$.

Value

diagnosticPlot only plots to a graphical device and doesn't return anything. Use the function quantileResidualTests in order to obtain the individual statistics.

Suggested packages

Install the suggested package "gsl" for faster evaluations in the cases of StMAR and G-StMAR models. For large StMAR and G-StMAR models with large data the calculations to obtain the individual statistics may take a significantly long time without the package "gsl".

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [**econ.EM**].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

[fitGSMAR](#), [GSMAR](#), [quantileResidualTests](#), [predict.gsmar](#)

Examples

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(data=VIX, p=1, M=3, params=params13, model="GMAR")
diagnosticPlot(gmar13)

# Restricted GMAR model: plot also the individual statistics with
# their approximate critical bounds using the given data
params12r <- c(1.4, 1.8, 0.88, 0.29, 3.18, 0.84)
gmar12r <- GSMAR(data=VIX, p=1, M=2, params=params12r, model="GMAR",
  restricted=TRUE)
diagnosticPlot(gmar12r, nlags=10, nsimu=1, plot_indstats=TRUE)

# StMAR model
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12 <- GSMAR(data=VIX, p=1, M=2, params=params12t, model="StMAR")
diagnosticPlot(stmar12)

# G-StMAR model (similar to the StMAR model above)
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(data=VIX, p=1, M=c(1, 1), params=params12gs,
  model="G-StMAR")
diagnosticPlot(gstmar12)

# Restricted G-StMAR-model
params13gsr <- c(1.3, 1, 1.4, 0.8, 0.4, 2, 0.2, 0.25, 0.15, 20)
gstmar13r <- GSMAR(data=VIX, p=1, M=c(2, 1), params=params13gsr,
  model="G-StMAR", restricted=TRUE)
diagnosticPlot(gstmar13r)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.85, 0.04, 0.3, 3.3, 0.77, 2.8, 0.77)
gmar22c <- GSMAR(data=VIX, p=2, M=2, params=params22c,
  model="GMAR", constraints=constraints)
diagnosticPlot(gmar22c)
```

```
# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
stmar32rc <- GSMAR(data=VIX, p=3, M=2, params=params32trc, model="StMAR",
  restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
diagnosticPlot(stmar32rc)
```

extractRegime	<i>Extract regime from a parameter vector</i>
---------------	---

Description

extractRegime extracts the specified regime from the GMAR, StMAR or G-StMAR model's parameter vector. Doesn't extract mixing weight parameter alpha.

Usage

```
extractRegime(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, regime)
```

Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.
For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.
For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.
For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.
For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.
regime	a positive integer in the interval [1, M] defining which regime should be extracted.

Value

Returns a numeric vector corresponding to the regime with...

For non-restricted models: For GMAR model: Size $(p+2x1)$ vector $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2)$.

For StMAR model: Size $(p+3x1)$ vector $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2, \nu_m)$.

For G-StMAR model: Same as GMAR for GMAR-components and same as StMAR for StMAR-components.

With linear constraints: Parameter vector as described above, but vector ϕ_m replaced with vector ψ_m that satisfies $\phi_m = R_m \psi_m$.

For restricted models: For GMAR model: Size $(2x1)$ vector $(\phi_{m,0}, \sigma_m^2)$.

For StMAR model: Size $(3x1)$ vector $(\phi_{m,0}, \sigma_m^2, \nu_m)$.

For G-StMAR model: Same as GMAR for GMAR-components and same as StMAR for StMAR-components.

With linear constraints: Parameter vector as described above.

fitGMAR	<i>fitGMAR is deprecated</i>
---------	------------------------------

Description

This function is deprecated! Use `fitGSMAR()` instead!

Usage

```
fitGMAR(...)
```

Arguments

... deprecated

fitGSMAR	<i>Estimate Gaussian or Student's t Mixture Autoregressive model</i>
----------	--

Description

`fitGSMAR` estimates GMAR, StMAR or G-StMAR model in two phases: in the first phase it uses genetic algorithm to find starting values for gradient based variable metric algorithm (also known as quasi-Newton method), which it then uses to finalize the estimation in the second phase. Parallel computing is used to perform multiple rounds of estimations in parallel.

Usage

```
fitGSMAR(data, p, M, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), nCalls = round(10 + 9 *
  log(sum(M))), nCores = min(nCalls, parallel::detectCores()), maxit = 100,
  printRes = TRUE, runTests = FALSE, ...)
```

Arguments

<code>data</code>	a numeric vector class 'ts' object containing the data. NA values are not supported.
<code>p</code>	a positive integer specifying the order of AR coefficients.
<code>M</code>	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.
conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$?
nCalls	a positive integer specifying how many rounds of estimation should be done. The estimation results may vary from round to round because of multimodality of the log-likelihood function and randomness associated with the genetic algorithm.
nCores	a positive integer specifying the number of cores to be used in the estimation process. Default is that the number of available cores is detected with <code>parallel::detectCores()</code> and all of them are used.
maxit	maximum number of iterations in the variable metric algorithm.
printRes	should the estimation results be printed?
runTests	should quantile residuals tests be performed after the estimation?
...	additional settings passed to the function <code>GAFit()</code> employing the genetic algorithm.

Details

Because of complexity and multimodality of the log-likelihood function, it's **not guaranteed** that the estimation algorithms will end up in the global maximum point. It's expected that most of the estimation rounds will end up in some local maximum point instead, and therefore a number of estimation rounds is required for reliable results. Because of the nature of the models, the estimation may fail particularly in the cases where the number of mixture components is chosen too large.

If the iteration limit in the variable metric algorithm (`maxit`) is reached, one can continue estimation by iterating more with the function `iterate_more()`.

The genetic algorithm is mostly based on the description by *Dorsey and Mayer (1995)*. It uses (slightly modified) individually adaptive crossover and mutation rates described by *Patnaik and Srinivas (1994)* and employs (50%) fitness inheritance discussed by *Smith, Dike and Stegmann (1995)*.

The variable metric algorithm (or quasi-Newton method) used in the second phase is implemented with function `optim` from the package `stats`.

Some mixture components of StMAR model may sometimes get very large degrees of freedom parameter estimates. Such estimates may, for example, cause computing the quantile residual tests to fail. However, such mixture components are very much similar to the components of GMAR model. It's hence advisable to further estimate a G-StMAR model by allowing the mixture components with large degrees of freedom estimates to be GMAR type.

Value

Returns an object of class 'gsmar' defining the estimated GMAR, StMAR or G-StMAR model. The returned object contains empirical mixing weights, quantile residuals and quantile residual test results if the tests were performed. In addition, the returned object contains the estimates and log-likelihood values from all the estimation rounds. The estimated parameter vector can be obtained at `gsmar$params` (and the corresponding approximate standard errors at `gsmar$std_errors`) and it is...

For non-restricted models: For GMAR model: Size $(M(p+3)-1 \times 1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m=(\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m=(\phi_{m,1}, \dots, \phi_{m,p})$, $m=1, \dots, M$.

For StMAR model: Size $(M(p+4)-1 \times 1)$ vector $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3)+M2-1 \times 1)$ vector $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m=R_m\psi_m$ for all $m=1, \dots, M$, where $\psi_m=(\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M+p-1 \times 1)$ vector $\theta=(\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi=(\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M+p-1 \times 1)$ vector $(\theta, \nu)=(\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M+M2+p-1 \times 1)$ vector $(\theta, \nu)=(\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi=R\psi$, where $\psi=(\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first **M1** components are *GMAR-type* and the rest **M2** components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

S3 methods

The following S3 methods are supported for class 'gsmar' objects: `print`, `summary`, `plot`, `logLik`, `residuals`.

Suggested packages

For faster evaluation of the quantile residuals of StMAR and G-StMAR models install the suggested package "gsl". Note that for large StMAR and G-StMAR models with large data the evaluations for the quantile residual tests may take significantly long time without the package "gsl".

References

- Dorsey R. E. and Mayer W. J. 1995. Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. *Journal of Business & Economic Statistics*, **13**, 53-66.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- Patnaik L.M. and Srinivas M. 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *Transactions on Systems, Man and Cybernetics* **24**, 656-667.
- Smith R.E., Dike B.A., Stegmann S.A. 1995. Fitness inheritance in genetic algorithms. *Proceedings of the 1995 ACM Symposium on Applied Computing*, 345-350.
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

[GSMAR](#), [iterate_more](#), [add_data](#), [swap_parametrization](#), [get_gradient](#), [simulateGSMAR](#), [predict.gsmar](#), [diagnosticPlot](#), [, quantileResidualTests](#)

Examples

```
# These are long running examples and use parallel computing

# GMAR model
fit12 <- fitGSMAR(VIX, 1, 2, runTests=TRUE)
fit12
summary(fit12)
plot(fit12)

# Restricted GMAR model
fit12r <- fitGSMAR(VIX, 1, 2, restricted=TRUE,
  parametrization="mean", nCalls=10)
fit12r
summary(fit12r)

# Non-mixture version of StMAR model
fit11t <- fitGSMAR(VIX, 1, 1, model="StMAR", nCores=1, nCalls=1)
fit11t

# StMAR model, 100 estimations rounds
fit12t <- fitGSMAR(VIX, 1, 2, model="StMAR", nCalls=100)
fit12t

# Restricted StMAR model (implied by the StMAR(1,2) model)
fit12tr <- fitGSMAR(VIX, 1, 2, model="StMAR", restricted=TRUE)
fit12tr
```

```

# G-StMAR model (implied by the StMAR(1,2) models), 100 estimation rounds
fit12gs <- fitGSMAR(VIX, 1, c(1, 1), model="G-StMAR", nCalls=100)
fit12gs

# Restricted G-StMAR model (implied by the previous StMAR and G-StMAR models)
fit12gsr <- fitGSMAR(VIX, 1, c(1, 1), model="G-StMAR", restricted=TRUE)
fit12gsr

# Fit GMAR model that is a mixture of AR(1) and such AR(3) model that the
# second AR coefficient is constrained to zero.
constraints <- list(matrix(c(1, 0, 0, 0, 0, 1), ncol=2), as.matrix(c(1, 0, 0)))
fit32c <- fitGSMAR(VIX, 3, 2, constraints=constraints)
fit32c

# Fit such constrained StMAR(3, 1) model that the second order AR coefficient
# is constrained to zero.
constraints <- list(matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
fit31tc <- fitGSMAR(VIX, 3, 1, model="StMAR", constraints=constraints)
fit31tc

# Fit such StMAR(3,2) model that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
fit32trc <- fitGSMAR(VIX, 3, 2, model="StMAR", restricted=TRUE,
                    constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
fit32trc

```

forecastGMAR

forecastGMAR is deprecated

Description

This function is deprecated! Use `predict.gsmar()` instead!

Usage

```
forecastGMAR(...)
```

Arguments

... deprecated

format_valuef	<i>Function factory for value formatting</i>
---------------	--

Description

format_valuef generates functions that format values so that they print out with the desired number of digits.

Usage

```
format_valuef(digits)
```

Arguments

digits	number of digits to use
--------	-------------------------

Value

returns a function that takes an atomic vector as argument and returns it formatted to character with digits decimals.

GAFit	<i>Genetic algorithm for preliminary estimation of GMAR, StMAR or G-StMAR model</i>
-------	---

Description

GAFit estimates specified GMAR, StMAR or G-StMAR model using genetic algorithm. It's designed to find starting values for gradient based methods.

Usage

```
GAFit(data, p, M, model = c("GMAR", "StMAR", "G-StMAR"), restricted = FALSE,
      constraints = NULL, parametrization = c("intercept", "mean"),
      conditional = TRUE, ngen, popsize, smartMu, meanscale, sigmascale,
      initpop = NULL, regime_force_scale = 1, red_criteria = c(0.05, 0.01),
      to_return = c("alt_ind", "best_ind"), minval)
```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components.

	<p>For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m=C_m\psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi=C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.</p>
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$?
conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
ngen	a positive integer specifying the number of generations to be ran through in the genetic algorithm. Default is $\min(400, \max(\text{round}(0.1 * \text{length}(\text{data})), 200))$.
popsize	a positive even integer specifying the population size in the genetic algorithm. Default is $10 * d$ where d is the number of parameters.
smartMu	a positive integer specifying the generation after which the random mutations in the genetic algorithm are "smart". This means that mutating individuals will mostly mutate fairly close (or partially close) to the best fitting individual so far. Default is $\min(100, \text{round}(0.5 * \text{ngen}))$.
meanscale	a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the μ_m mean-parameters are generated in random mutations in the genetic algorithm. Default is $\text{c}(\text{mean}(\text{data}), \text{sd}(\text{data}))$. Note that the genetic algorithm optimizes with mean-parametrization even when <code>parametrization="intercept"</code> , but input (in <code>initpop</code>) and output (return value) parameter vectors may be intercept-parametrized.
sigmascale	a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters are generated in the random mutations in the genetic algorithm. Default is <code>var(stats::ar(data, order.max=10)\$resid, na.rm=TRUE)</code> .
initpop	a list of parameter vectors from which the initial population of the genetic algorithm will be generated from. The parameter vectors should be of form...

For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices \mathbf{R} that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix R that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and v a degrees of freedom parameter. Note that in the case $\mathbf{M=1}$ the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2. If not specified (or FALSE as is default), the initial population will be drawn randomly.

regime_force_scale

a non-negative real number specifying how much should natural selection favour individuals with less redundant regimes (see red_criteria). Set to zero for no favouring or large number for heavy favouring. Without any favouring the genetic algorithm gets more often stuck in an area of the parameter space where some regimes are wasted, but with too much favouring the best genes might never mix into the population and the algorithm might converge poorly. Default is 1 and it gives $2x$ larger surviving probabilities for individuals with no wasted regimes compared to individuals with one wasted regime. Number 2 would give $3x$ larger probabilities etc.

red_criteria

a length 2 numeric vector specifying the criteria that is used to determine whether a regime is redundant or not. Any regime m which satisfies $\text{sum}(\text{mixingWeights}[,m]) > \text{red_criteria}[1]$ will be considered "redundant". One should be careful when adjusting this argument.

to_return

should the genetic algorithm return the best fitting individual which has as much "non-redundant" regimes as possible ("alt_ind") or the individual which has the highest log-likelihood in general ("best_ind"), but might have more wasted regimes?

minval

a real number defining the minimum value of the log-likelihood function that will be considered. Values smaller than this will be treated as they were minval and the corresponding individuals will never survive. The default is $-(10^{\text{ceiling}(\log_{10}(\text{length}(\text{data})))})$ and one should be very careful if adjusting this.

Details

The genetic algorithm is mostly based on the description by *Dorsey and Mayer (1995)*. It uses (slightly modified) individually adaptive crossover and mutation rates described by *Patnaik and*

Srinivas (1994) and employs (50%) fitness inheritance discussed by *Smith, Dike and Stegmann (1995)*.

Value

Returns estimated parameter vector described in `initpop`.

References

- Dorsey R. E. and Mayer W. J. 1995. Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. *Journal of Business & Economic Statistics*, **13**, 53-66.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- Patnaik L.M. and Srinivas M. 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *Transactions on Systems, Man and Cybernetics* **24**, 656-667.
- Smith R.E., Dike B.A., Stegmann S.A. 1995. Fitness inheritance in genetic algorithms. *Proceedings of the 1995 ACM Symposium on Applied Computing*, 345-350.
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

getOmega

Generate covariance matrix Omega for quantile residual tests

Description

getOmega generates the covariance matrix Omega used in the quantile residual tests.

Usage

```
getOmega(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, parametrization = c("intercept",
  "mean"), g, dim_g)
```

Arguments

- | | |
|------|--|
| data | a numeric vector class 'ts' object containing the data. NA values are not supported. |
| p | a positive integer specifying the order of AR coefficients. |
| M | For GMAR and StMAR models: a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$. |

params

a real valued parameter vector specifying the model.

For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model

is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted

a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

constraints

specifies linear constraints applied to the autoregressive parameters.

For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

parametrization

is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$?

g

a function specifying the transformation.

dim_g

output dimension of the transformation g .

Details

This function is used for quantile residuals tests in `quantileResidualTests`.

Value

Returns size (dim_gxdim_g) covariance matrix Omega.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

get_ar_roots	<i>Calculate absolute values of the roots of the AR characteristic polynomials</i>
--------------	--

Description

get_ar_roots calculates absolute values of the roots of the AR characteristic polynomials for each component.

Usage

```
get_ar_roots(gsmar)
```

Arguments

gsmar object of class 'gsmar', generated by function fitGSMAR() or GSMAR().

Value

Returns a list with M elements each containing the absolute values of the roots of the AR characteristic polynomial corresponding to each mixture component.

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

Examples

```
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(data=VIX, p=1, M=3, params=params13, model="GMAR")
get_ar_roots(gmar13)
```

get_IC	<i>Calculate AIC, HQIC and BIC</i>
--------	------------------------------------

Description

get_IC calculates AIC, HQIC and BIC

Usage

```
get_IC(loglik, npars, obs)
```

Arguments

loglik	log-likelihood value
npars	number of (freely estimated) parameters in the model
obs	numbers of observations with starting values excluded for conditional models.

Value

Returns a data frame containing the information criteria values.

get_regime_means	<i>Calculate and return regime means μ_m</i>
------------------	---

Description

get_regime_means calculates regime means $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ for the given GMAR, StMAR or G-StMAR model

Usage

```
get_regime_means(gsmar)
```

Arguments

gsmar	object of class 'gsmar', generated by function fitGSMAR() or GSMAR().
-------	---

Value

Returns a length M vector containing regime mean μ_m in the m:th column, $m = 1, \dots, M$.

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

Examples

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(data=VIX, p=1, M=3, params=params13, model="GMAR")
get_regime_means(gmar13)

# StMAR model
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12t <- GSMAR(data=VIX, p=1, M=2, params=params12t, model="StMAR")
get_regime_means(stmar12t)

# G-StMAR model (similar to the StMAR model above)
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(data=VIX, p=1, M=c(1, 1), params=params12gs,
  model="G-StMAR")
get_regime_means(gstmar12)
```

GSMAR

Create object of class 'gsmar' defining a GMAR, StMAR or G-StMAR model

Description

GSMAR creates an S3 object of class 'gsmar' that defines a GMAR, StMAR or G-StMAR model.

Usage

```
GSMAR(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), calc_qresiduals,
  calc_std_errors = FALSE)

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

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

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

## S3 method for class 'gsmar'
plot(x, ...)

## S3 method for class 'gsmar'
print(x, ..., digits = 2, summary_print = FALSE)
```

Arguments

data a numeric vector class 'ts' object containing the data. NA values are not supported.

p a positive integer specifying the order of AR coefficients.

M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is $M=M1+M2$.

params a real valued parameter vector specifying the model.

For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If `parameterization="mean"` just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.
conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$?
calc_qresiduals	should quantile residuals be calculated? Default is TRUE if the model contains data.
calc_std_errors	should approximate standard errors be calculated?
object	object of class 'gsmar' generated by fitGSMAR() or GSMAR().
...	graphical parameters passed to ts.plot.
x	object of class 'gsmar' generated by fitGSMAR() or GSMAR().
digits	number of digits to be printed
summary_print	if set to TRUE then the print will include approximate standard errors, log-likelihood and information criteria values. Supported only for models with data.

Details

Models without data can be created.

Methods (by generic)

- logLik: Log-likelihood method
- residuals: residuals method to extract multivariate quantile residuals
- summary: summary method, standard errors in brackets
- plot: plot method for class 'gsmar'
- print: print method

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

[fitGSMAR](#), [iterate_more](#), [add_data](#), [swap_parametrization](#), [get_gradient](#), [simulateGSMAR](#), [predict.gsmar](#)

Examples

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(data=VIX, p=1, M=3, params=params13, model="GMAR")
gmar13
```

```
# Restricted GMAR model
params12r <- c(1.4, 1.8, 0.88, 0.29, 3.18, 0.84)
gmar12r <- GSMAR(data=VIX, p=1, M=2, params=params12r, model="GMAR",
  restricted=TRUE)
gmar12r
```

```
# StMAR model, without data
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12t <- GSMAR(VIX, p=1, M=2, params=params12t, model="StMAR")
stmar12t
```

```
# G-StMAR model (similar to the StMAR model above)
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(data=VIX, p=1, M=c(1, 1), params=params12gs,
  model="G-StMAR")
gstmar12
```

```
# Restricted G-StMAR-model
params13gsr <- c(1.3, 1, 1.4, 0.8, 0.4, 2, 0.2, 0.25, 0.15, 20)
gstmar13r <- GSMAR(data=VIX, p=1, M=c(2, 1), params=params13gsr,
  model="G-StMAR", restricted=TRUE)
diagnosticPlot(gstmar13r)
```

```
# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.85, 0.04, 0.3, 3.3, 0.77, 2.8, 0.77)
gmar22c <- GSMAR(data=VIX, p=2, M=2, params=params22c,
  model="GMAR", constraints=constraints)
gmar22c
```

```
# Such StMAR(3,2) that the AR coefficients are restricted to be
```

```
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
stmar32rc <- GSMAR(data=VIX, p=3, M=2, params=params32trc, model="StMAR",
  restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
stmar32rc

# Mixture version of Heterogeneous Autoregressive (HAR) model (without data)
paramsHAR <- c(1, 0.1, 0.2, 0.3, 1, 2, 0.15, 0.25, 0.35, 2, 0.55)
r1 = c(1, rep(0, 21)); r2 = c(rep(0.2, 5), rep(0, 17)); r3 = rep(1/22, 22)
R0 = cbind(r1, r2, r3)
mixhar <- GSMAR(p=22, M=2, params=paramsHAR, model="GMAR", constraints=list(R0, R0))
mixhar
```

isStationary	<i>Check the stationary condition of specified GMAR, StMAR or G-StMAR model.</i>
--------------	--

Description

isStationary checks the stationarity condition of the specified GMAR, StMAR or G-StMAR model.

Usage

```
isStationary(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

- | | |
|--------|--|
| p | a positive integer specifying the order of AR coefficients. |
| M | <p>For GMAR and StMAR models: a positive integer specifying the number of mixture components.</p> <p>For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.</p> |
| params | <p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> |

For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

Details

This function uses numerical approximations and it will falsely return FALSE for stationary models when the stationarity condition is really close to break.

Value

Returns TRUE or FALSE accordingly.

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

Examples

```

# GMAR model
params22 <- c(0.4, 0.39, 0.6, 0.3, 0.4, 0.1, 0.6, 0.3, 0.8)
isStationary(2, 2, params22)

# StMAR model
params12t <- c(-0.3, 1, 0.9, 0.1, 0.8, 0.6, 0.7, 10, 12)
isStationary(1, 2, params12t, model="StMAR")

# G-StMAR model
params12gs <- c(1, 0.1, 1, 2, 0.2, 2, 0.8, 20)
isStationary(1, c(1, 1), params12gs, model="G-StMAR")

# Restricted GMAR model
params13r <- c(0.1, 0.2, 0.3, -0.99, 0.1, 0.2, 0.3, 0.5, 0.3)
isStationary(1, 3, params13r, restricted=TRUE)

# Restricted StMAR model
params22tr <- c(-0.1, -0.2, 0.01, 0.99, 0.3, 0.4, 0.9, 3, 13)
isStationary(2, 2, params22tr, model="StMAR", restricted=TRUE)

# Restricted G-StMAR model
params13gsr <- c(1, 2, 3, -0.99, 1, 2, 3, 0.5, 0.4, 20, 30)
isStationary(1, c(1, 2), params13gsr, model="G-StMAR", restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.8, 0.2, 0.3, 3.3, 0.7, 3, 0.8)
isStationary(2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be the
# same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(1, 2, 0.8, -0.3, 1, 2, 0.7, 11, 12)
isStationary(3, 2, params32trc, model="StMAR", restricted=TRUE,
             constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))

```

isStationary_int	<i>Check the stationary and identification conditions of specified GMAR, StMAR or G-StMAR model.</i>
------------------	--

Description

isStationary_int checks the stationary condition and isIdentifiable checks the identification conditions of the specified GMAR, StMAR or G-StMAR model.

Usage

```
isStationary_int(p, M, params, restricted = FALSE)
```

```
isIdentifiable(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2×1) vector specifying the number of *GMAR-type* components $M1$ in the first element and *StMAR-type* components $M2$ in the second. The total number of mixture components is $M=M1+M2$.
- params** a real valued parameter vector specifying the model.
For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.
For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.
For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
- Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. In the **G-StMAR** model the first $M1$ components are *GMAR-type* and the rest $M2$ components are *StMAR-type*. Note that in the case $\mathbf{M=1}$ the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.
- restricted** a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first $M1$ components are *GMAR-type* and the rest $M2$ components are *StMAR-type*.
- constraints** specifies linear constraints applied to the autoregressive parameters.
For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.
For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.
- Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

Details

These functions don't support models parametrized with general linear constraints.

Value

Returns TRUE or FALSE accordingly.

Warning

These functions don't have any argument checks!

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

iterate_more	<i>Maximum likelihood estimation of GMAR, StMAR or G-StMAR model with preliminary estimates</i>
--------------	---

Description

iterate_more uses variable metric algorithm to finalize maximum likelihood estimation of GMAR, StMAR or G-StMAR model (object of class 'gsmarar') which already has preliminary estimates.

Usage

```
iterate_more(gsmar, maxit = 100)
```

Arguments

gsmar	object of class 'gsmar', generated by function fitGSMAR() or GSMAR().
maxit	maximum number of iterations in the variable metric algorithm.

Details

The main purpose of iterate_more() is to provide a simple and convenient tool to finalize the estimation when the maximum number of iterations is reached when estimating a model with the main estimation function fitGSMAR(). It's just a simple wrapper around function optim() from the package stats and GSMAR() from the package uGMAR.

Value

Returns an object of class 'gsmar' defining the estimated model. Can be used to work with other functions provided in uGMAR.

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

`fitGSMAR()`, `GSMAR()`, `optim()`

Examples

```
# Estimate GMAR model with only 50 generations of genetic algorithm and
# only 1 iteration in variable metric algorithm
fit12 <- fitGSMAR(VIX, 1, 2, maxit=1, ngen=50)
fit12

# Iterate more since iteration limit was reached
fit12 <- iterate_more(fit12)
fit12
```

loglikelihood

Compute the log-likelihood of GMAR, StMAR or G-StMAR model

Description

`loglikelihood` computes the log-likelihood value of the specified GMAR, StMAR or G-StMAR model. Exists for convenience if one wants to for example plot profile log-likelihoods or employ other estimation algorithms. Use `minval` to control what happens when the parameter vector is outside the parameter space.

Usage

```
loglikelihood(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), returnTerms = FALSE,
  minval = NA)
```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<p>For GMAR and StMAR models: a positive integer specifying the number of mixture components.</p> <p>For G-StMAR model: a size (2×1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size $(p \times q_m)$ constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size $(p \times q)$ constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p>

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$?
returnTerms	should the terms $l_t : t = 1, \dots, T$ in the log-likelihood function (see <i>KMS 2015, eq.(13)</i>) be returned instead of the log-likelihood value?
minval	this will be returned when the parameter vector is outside the parameter space.

Value

Returns the log-likelihood value or the terms described in returnTerms.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [**econ.EM**].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

[fitGSMAR](#), [GSMAR](#), [quantileResiduals](#), [mixingWeights](#), [calc_gradient](#)

Examples

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
loglikelihood(VIX, 1, 3, params13)

# Restricted GMAR model, outside parameter space
params12r <- c(1.4, 1.8, 1.88, 0.29, 3.18, 0.84)
loglikelihood(VIX, 1, 2, params12r, restricted=TRUE, minval=-9999)

# StMAR model
params12t <- c(1.9, 0.85, 1.16, 1.22, 0.89, 0.13, 0.64, 3.1, 7.0)
loglikelihood(VIX, 1, 2, params12t, model="StMAR")

# Non-mixture version of StMAR model, outside parameter space
params11t <- c(0.76, 0.93, 1.35, 1.9)
```

```

loglikelihood(VIX, 1, 1, params11t, model="StMAR", minval="Hello")

# G-StMAR model
params12gs <- c(1.2, 0.8, 0.6, 1.3, 0.6, 1.1, 0.6, 3)
loglikelihood(VIX, 1, c(1,1), params12gs, model="G-StMAR")

# Restricted G-StMAR model
params13gsr <- c(1.3, 2.2, 1.4, 0.8, 2.4, 4.6, 0.4, 0.25, 0.15, 20)
loglikelihood(VIX, 1, c(2, 1), params13gsr, model="G-StMAR", restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.85, 0.04, 0.3, 3.3, 0.77, 2.8, 0.77)
loglikelihood(VIX, 2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
loglikelihood(VIX, 3, 2, params32trc, model="StMAR", restricted=TRUE,
              constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))

```

loglikelihood_int

Compute the log-likelihood of GMAR, StMAR or G-StMAR model

Description

loglikelihood_int computes the log-likelihood value of the specified GMAR, StMAR or G-StMAR model for the given data.

Usage

```

loglikelihood_int(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), boundaries = TRUE,
  checks = TRUE, to_return = c("loglik", "mw", "mw_tplus1", "loglik_and_mw",
  "terms"), minval)

```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$.

params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices \mathbf{C} that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.</p>
conditional parametrization	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
boundaries	<p>is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$?</p> <p>a logical argument. If TRUE then loglikelihood returns minval if...</p> <ul style="list-style-type: none"> • any component variance is not larger than zero,

- any parametrized mixing weight $\alpha_1, \dots, \alpha_{M-1}$ is not larger than zero,
- sum of the parametrized mixing weights is not smaller than one,
- if the model is not stationary,
- or if `model=="StMAR"` or `model=="G-StMAR"` and any degrees of freedom parameter ν_m is not larger than two.

Argument `minval` will be used only if `boundaries==TRUE`.

<code>checks</code>	an (optional) logical argument defining whether argument checks are made. If <code>FALSE</code> then no argument checks such as stationary checks etc are made. The default is <code>TRUE</code> .
<code>to_return</code>	should the returned object be the log-likelihood value, mixing weights, mixing weights including value for $\alpha_{m,T+1}$, a list containing log-likelihood value and mixing weights or the terms $l_t : t = 1, \dots, T$ in the log-likelihood function (see <i>KMS 2015, eq.(13)</i>)? Default is the log-likelihood value (" <code>loglik</code> ").
<code>minval</code>	this will be returned when the parameter vector is outside the parameter space.

Value

By default: log-likelihood value of the specified model,

If `to_return=="mw"`: a size $((n_obs-p) \times M)$ matrix containing the mixing weights: for m :th component in m :th column.

If `to_return=="mw_tplus1"`: a size $((n_obs-p+1) \times M)$ matrix containing the mixing weights: for m :th component in m :th column. The last row is for $\alpha_{m,T+1}$.

If `to_return=="terms"`: a size $((n_obs-p) \times 1)$ numeric vector containing the terms l_t .

if `to_return=="loglik_and_mw"`: a list of two elements. The first element contains the log-likelihood value and the second element contains the mixing weights.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

mixingWeights	Calculate mixing weights of GMAR, StMAR or G-StMAR model
---------------	--

Description

mixingWeights calculates the mixing weights of the specified GMAR, StMAR or G-StMAR model and returns them as a matrix.

Usage

```
mixingWeights(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, parametrization = c("intercept",
  "mean"))
```

Arguments

data a numeric vector class 'ts' object containing the data. NA values are not supported.

p a positive integer specifying the order of AR coefficients.

M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

params a real valued parameter vector specifying the model.

For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest

M2 components are *StMAR-type*. Note that in the case $\mathbf{M=1}$ the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$?

Value

- If to_return=="mw":** a size $((n_obs-p) \times M)$ matrix containing the mixing weights: for m:th component in m:th column.
- If to_return=="mw_tplus1":** a size $((n_obs-p+1) \times M)$ matrix containing the mixing weights: for m:th component in m:th column. The last row is for $\alpha_{m,T+1}$.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [**econ.EM**].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

Examples

```
# GMAR model
params12 <- c(1.1, 0.9, 0.3, 4.5, 0.7, 3.2, 0.8)
mw12 <- mixingWeights(VIX, 1, 2, params12)
```

```

# Restricted GMAR model
params12r <- c(1.4, 1.8, 0.9, 0.3, 3.2, 0.8)
mw12r <- mixingWeights(VIX, 1, 2, params12r, restricted=TRUE)

# StMAR model
params12t <- c(1.1, 0.9, 0.3, 4.5, 0.7, 3.2, 0.8, 5, 8)
mw12t <- mixingWeights(VIX, 1, 2, params12t, model="StMAR")

# Non-mixture version of StMAR model
params11t <- c(0.76, 0.93, 1.35, 2.4)
mw11t <- mixingWeights(VIX, 1, 1, params11t, model="StMAR")

# G-StMAR model
params12gs <- c(1.2, 0.8, 0.6, 1.3, 0.6, 1.1, 0.6, 3)
mw12gs <- mixingWeights(VIX, 1, c(1,1), params12gs, model="G-StMAR")

# Restricted G-StMAR model
params13gsr <- c(1.3, 2.2, 1.4, 0.8, 2.4, 4.6, 0.4, 0.25, 0.15, 20)
mw13gsr <- mixingWeights(VIX, 1, c(2,1), params13gsr, model="G-StMAR", restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.8, 0.1, 0.3, 3.3, 0.8, 2.8, 0.8)
mw22c <- mixingWeights(VIX, 2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
mw32trc <- mixingWeights(VIX, 3, 2, params32trc, model="StMAR",
                        restricted=TRUE,
                        constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))

```

mixingWeights_int *Calculate mixing weights of GMAR, StMAR or G-StMAR model*

Description

mixingWeights_int calculates the mixing weights of the specified GMAR, StMAR or G-StMAR model and returns them as a matrix.

Usage

```

mixingWeights_int(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, parametrization = c("intercept",
  "mean"), checks = TRUE, to_return = c("mw", "mw_tplus1"))

```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<p>For GMAR and StMAR models: a positive integer specifying the number of mixture components.</p> <p>For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3)-1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4)-1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3)+M2-1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M+p-1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M+p-1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M+M2+p-1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p>

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$?
checks	an (optional) logical argument defining whether argument checks are made. If FALSE then no argument checks such as stationary checks etc are made. The default is TRUE.
to_return	should the returned object the mixing weights or mixing weights ("mw") including value for $\alpha_{m,T+1}$ ("mw_tplus1")?

Value

If to_return=="mw": a size $((n_obs-p) \times M)$ matrix containing the mixing weights: for m :th component in m :th column.

If to_return=="mw_tplus1": a size $((n_obs-p+1) \times M)$ matrix containing the mixing weights: for m :th component in m :th column. The last row is for $\alpha_{m,T+1}$.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [**econ.EM**].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

nParams

Calculate the number of parameters

Description

nParams calculates the number of parameters that should be in the parameter vector.

Usage

```
nParams(p, M, model = c("GMAR", "StMAR", "G-StMAR"), restricted = FALSE,
constraints = NULL)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m=C_m\psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi=C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = \psi_1, \dots, \psi_q$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.

Value

Returns the number of parameters.

parameterChecks	<i>Check the parameter vector is specified correctly</i>
-----------------	--

Description

parameterChecks checks dimension, restrictions and stationarity of the given parameters of GMAR, StMAR or G-StMAR model. Throws an error if any check fails. Does NOT consider the identifiability condition!

Usage

```
parameterChecks(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
params	a real valued parameter vector specifying the model. For GMAR model: Size $(M(p+3)-1 \times 1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m=(\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m=(\phi_{m,1}, \dots, \phi_{m,p})$, $m=1, \dots, M$. For StMAR model: Size $(M(p+4)-1 \times 1)$ vector $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$. For G-StMAR model: Size $(M(p+3)+M2-1 \times 1)$ vector $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size $(p \times q_m)$ constraint matrices C_m of full column rank satisfying $\phi_m=C_m\psi_m$ for all $m=1, \dots, M$, where $\phi_m=(\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m=(\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size $(p \times q)$ constraint matrix C of full column rank satisfying $\phi=C\psi$, where $\phi=(\phi_1, \dots, \phi_p)$ and $\psi=(\psi_1, \dots, \psi_q)$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.

Value

Throws an informative error if any check fails. Doesn't return anything.

pick_alphas	<i>Pick mixing weights parameters from parameter vector</i>
-------------	---

Description

pick_alphas picks and returns the mixing weights parameters (including the non-parametrized one for the last component) from parameter vector.

Usage

```
pick_alphas(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	<p>For GMAR and StMAR models: a positive integer specifying the number of mixture components.</p> <p>For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.</p>

Value

Returns a vector of length M containing the mixing weights parameters α_m .

pick_dfs

Pick degrees of freedom parameters from parameter vector

Description

pick_dfs picks and returns the degrees of freedom parameters from parameter vector.

Usage

```
pick_dfs(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"))
```

Arguments

p a positive integer specifying the order of AR coefficients.

M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is $M=M1+M2$.

params a real valued parameter vector specifying the model.

For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

Value

Returns a vector of length M or M2 containing the degrees of freedom parameters

pick_pars	<i>Pick ϕ_0/μ, AR-coefficients and variance parameters from parameter vector</i>
-----------	--

Description

pick_pars picks ϕ_0/μ , ar-coefficient and variance parameters from parameter vector

Usage

```
pick_pars(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

p a positive integer specifying the order of AR coefficients.

M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is $M=M1+M2$.

params a real valued parameter vector specifying the model.
For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.
For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.
For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.

Value

Returns a $(Mx(p + 2))$ matrix containing the parameters, column for each component. First row for ϕ_0/μ depending on the parametrization, second row for ϕ_1 , second last row for ϕ_p and last row for σ^2 .

pick_phi0	<i>Pick phi0 or mean parameters from parameter vector</i>
-----------	---

Description

pick_phi0 picks and returns the phi0 or mean parameters from parameter vector.

Usage

```
pick_phi0(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

p a positive integer specifying the order of AR coefficients.
M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

	<p>For G-StMAR model: a size (2×1) vector specifying the number of <i>GMAR-type</i> components $M1$ in the first element and <i>StMAR-type</i> components $M2$ in the second. The total number of mixture components is $M=M1+M2$.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first $M1$ components are <i>GMAR-type</i> and the rest $M2$ components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first $M1$ components are <i>GMAR-type</i> and the rest $M2$ components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.</p>

Value

Returns a vector of length M containing the phi0 or mean parameters depending on parametrization.

plot.gsmarpred *plot method for class 'gsmarpred' objects*

Description

plot.gsmarpred is plot method for gsmarpred objects

Usage

```
## S3 method for class 'gsmarpred'
plot(x, ..., nt, add_grid = TRUE)
```

Arguments

x	object of class 'gsmarpred' generated by predict.gsmar().
...	arguments passed to grid()
nt	a positive integer specifying the number of observations to be plotted along with the prediction. Default is round(length(data)*0.2).
add_grid	should grid be added to the plots?

Details

This method is used plot forecasts of gsmar processes

Examples

```
# GMAR-model
params12 <- c(1.12321, 0.90557, 0.29288, 4.53131,
             0.70309, 3.21154, 0.83888)
gmar12 <- GSMAR(VIX, 1, 2, params12)
pred <- predict(gmar12, n_ahead=10, plotRes=FALSE)
plot(pred, nt=50)
```

plot.qrtest *Quantile residual tests for GMAR, StMAR or G-StMAR model*

Description

quantileResidualTests performs quantile residual tests for GMAR, StMAR or G-StMAR model, testing normality, autocorrelation and conditional heteroscedasticity.

Usage

```
## S3 method for class 'qrtest'
plot(x, ...)

## S3 method for class 'qrtest'
print(x, ..., digits = 3)

quantileResidualTests(gsmar, lagsAC = c(1, 2, 5, 10), lagsCH = lagsAC,
  nsimu = 2000, printRes = TRUE)
```

Arguments

x	object of class 'qrtest' generated by the function <code>quantileResidualTests()</code> .
...	graphical parameters passed to segments in <code>plot.qrtest</code> . Currently not used in <code>print.qrtest</code>
digits	number of decimals to print
gsmar	object of class 'gsmar', generated by function <code>fitGSMAR()</code> or <code>GSMAR()</code> .
lagsAC	a numeric vector of positive integers specifying the lags for which autocorrelation is tested.
lagsCH	a numeric vector of positive integers specifying the lags for which conditional heteroscedasticity is tested.
nsimu	a positive integer specifying to how many simulated observations the covariance matrix Omega should be based on. If smaller than data size, then omega will be based on the given data.
printRes	a logical argument defining whether results should be printed or not.

Details

For details about the quantile residual tests see the cited article by *Kalliovirta (2012)*.

Value

Returns an object of class 'qrtest' containing the test results in data frames. In the cases of autocorrelation and conditional heteroscedasticity tests, the returned object also contains the associated individual statistics and their standard errors, discussed by *Kalliovirta (2012)* at the pages 369-370.

Methods (by generic)

- `plot`: plot p-values of the autocorrelation and conditional heteroskedasticity tests.
- `print`: print method for class 'qrtest'

Suggested packages

Install the suggested package "gsl" for faster evaluations in the cases of StMAR and G-StMAR models. For large StMAR and G-StMAR models with large data the evaluations may take significantly long time without the package "gsl".

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

[fitGSMAR](#), [GSMAR](#), [diagnosticPlot](#), [predict.gsmar](#),

Examples

```
# GMAR model
params12 <- c(1.12, 0.91, 0.29, 4.53, 0.70, 3.21, 0.84)
gmar12 <- GSMAR(VIX, 1, 2, params12)
qrtest12 <- quantileResidualTests(gmar12)
plot(qrtest12)

# Restricted GMAR model
params12r <- c(1.4, 1.8, 0.88, 0.29, 3.18, 0.84)
gmar12r <- GSMAR(data=VIX, p=1, M=2, params=params12r, model="GMAR",
  restricted=TRUE)
qrtest12r <- quantileResidualTests(gmar12r, lagsAC=1:10, nsimu=1)
plot(qrtest12r)

# StMAR model
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12 <- GSMAR(data=VIX, p=1, M=2, params=params12t, model="StMAR")
quantileResidualTests(stmar12, lagsAC=c(1, 2, 5), nsimu=1)

# G-StMAR model
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(data=VIX, p=1, M=c(1, 1), params=params12gs,
  model="G-StMAR")
quantileResidualTests(gstmar12, lagsAC=c(1, 3), lagsCH=1:2, nsimu=1)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
stmar32rc <- GSMAR(data=VIX, p=3, M=2, params=params32trc, model="StMAR",
```

```
restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
quantileResidualTests(stmar32rc, lagsAC=c(1, 3), nsimu=1)
```

plotGMAR	<i>plotGMAR is deprecated</i>
----------	-------------------------------

Description

This function is deprecated! Use `diagnosticPlot()` instead!

Usage

```
plotGMAR(...)
```

Arguments

... deprecated

predict.gsmar	<i>Forecast GMAR, StMAR or G-StMAR process</i>
---------------	--

Description

`forecastGSMAR` forecasts the specified GMAR, StMAR or G-StMAR process by using the given data to simulate its possible future values. For one-step forecasts using the exact formula of conditional mean is supported.

Usage

```
## S3 method for class 'gsmar'
predict(object, ..., n_ahead, nsimu = 5000, ci = c(0.95,
  0.8), pred_type = c("mean", "median", "cond_mean"),
  ci_type = c("two-sided", "upper", "lower", "none"), nt, plotRes = TRUE)
```

Arguments

object	object of class 'gsmar', generated by function <code>fitGSMAR()</code> or <code>GSMAR()</code> .
...	additional arguments passed to <code>grid()</code> (ignored if <code>plot_res==FALSE</code>).
n_ahead	a positive integer specifying how many steps in the future should be forecasted.
nsimu	a positive integer specifying to how many simulations the forecast should be based on.
ci	a numeric vector specifying the confidence levels of the confidence intervals.

pred_type	should the prediction be based on sample "mean" or "median"? Or should it be one-step-ahead forecast based on conditional mean ("cond_mean")? Confidence intervals won't be calculated if conditional mean is used.
ci_type	should the confidence intervals be "two-sided", "upper" or "lower"?
nt	a positive integer specifying the number of observations to be plotted along with the prediction. Default is round(length(data)*0.2).
plotRes	a logical argument defining whether the forecast should be plotted or not.

Details

forecastGSMAR uses the last p values of the data to simulate n_{simu} possible future values for each step. The best prediction is then obtained by calculating the sample median (or mean) of each step and the confidence intervals are obtained from the empirical fractiles.

Value

Returns a data frame containing the empirical best prediction and confidence intervals accordingly to ci. Or if pred_type=="cond_mean" returns the optimal prediction as (1x1) numeric vector.

See Also

[fitGSMAR](#), [GSMAR](#), [quantileResidualTests](#), [diagnosticPlot](#)

Examples

```
# GMAR model
params12 <- c(1.12, 0.91, 0.29, 4.53, 0.70, 3.21, 0.84)
gmar12 <- GSMAR(VIX, 1, 2, params12)
pred12 <- predict(gmar12, n_ahead=10)
pred12

# Restricted GMAR model, one-step conditional mean prediction
params12r <- c(1.4, 1.8, 0.88, 0.29, 3.18, 0.84)
gmar12r <- GSMAR(data=VIX, p=1, M=2, params=params12r, model="GMAR",
  restricted=TRUE)
pred12r <- predict(gmar12r, pred_type="cond_mean", plotRes=FALSE)
pred12r

# StMAR model, upper confidence intervals
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12 <- GSMAR(data=VIX, p=1, M=2, params=params12t, model="StMAR")
predict(stmar12, n_ahead=10, ci_type="upper", ci=c(0.99, 0.95, 0.9))

# G-StMAR model, no confidence intervals
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(data=VIX, p=1, M=c(1, 1), params=params12gs,
  model="G-StMAR")
pred12gs <- predict(gstmar12, n_ahead=10, pred_type="median",
  ci_type="none", plotRes=FALSE)
pred12gs
```

```

plot(pred12gs)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.85, 0.04, 0.3, 3.3, 0.77, 2.8, 0.77)
gmar22c <- GSMAR(data=VIX, p=2, M=2, params=params22c,
  model="GMAR", constraints=constraints)
predict(gmar22c, n_ahead=5, nsimu=10000, nt=10)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
stmar32rc <- GSMAR(data=VIX, p=3, M=2, params=params32trc, model="StMAR",
  restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
predict(stmar32rc, n_ahead=3, ci_type="lower")

```

```
print.gsmarpred      print method for class 'gsmarpred' objects
```

Description

print.gsmarpred is print method for object generated by predict.gsmar().

Usage

```
## S3 method for class 'gsmarpred'
print(x, ..., digits = 2)
```

Arguments

x	object of class 'gsmarpred' generated by predict.gsmar().
...	currently not used.
digits	number of decimals to print

Examples

```

# GMAR-model
params12 <- c(1.12321, 0.90557, 0.29288, 4.53131,
  0.70309, 3.21154, 0.83888)
gmar12 <- GSMAR(VIX, 1, 2, params12)
pred <- predict(gmar12, n_ahead=10, plotRes=FALSE)
pred
print(pred, digits=3)

```

```
print.gsmarsum      Print method from objects of class 'gsmarsum'
```

Description

print.gsmarsum is a print method for object 'gsmarsum' generated by summary.gsmar(). Approximate standard errors are printed in brackets.

Usage

```
## S3 method for class 'gsmarsum'
print(x, ..., digits = 2)
```

Arguments

x	object of class 'gsmarsum' generated by summary.gsmar().
...	currently not used.
digits	number of digits to be printed

Examples

```
# GMAR model
fit12 <- fitGSMAR(VIX, 1, 2, nCalls=4)
gsmarsum12 <- summary(fit12)
gsmarsum12
print(gsmarsum12, digits=4)
```

```
quantileResiduals      Compute quantile residuals of GMAR, StMAR or G-StMAR model
```

Description

quantileResiduals computes the quantile residuals of the specified GMAR, StMAR or G-StMAR model.

Usage

```
quantileResiduals(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, parametrization = c("intercept",
  "mean"))
```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<p>For GMAR and StMAR models: a positive integer specifying the number of mixture components.</p> <p>For G-StMAR model: a size (2×1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p>

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

parametrization

is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$?

Value

Returns a $(Tx1)$ numeric vector containing the quantile residuals associated with the specified GMAR, StMAR or G-StMAR model.

Suggested packages

Install the suggested package "gsl" for faster evaluation of the quantile residuals of StMAR and G-StMAR models.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

Examples

```
# GMAR model
params12 <- c(1.12, 0.9, 0.29, 4.53, 0.7, 3.22, 0.84)
qr12 <- quantileResiduals(VIX, 1, 2, params12)

# Restricted GMAR model
params12r <- c(1.4, 1.8, 0.88, 0.29, 3.18, 0.84)
qr12r <- quantileResiduals(VIX, 1, 2, params12r, restricted=TRUE)

# StMAR model
params12t <- c(1.1, 0.9, 0.3, 4.5, 0.7, 3.2, 0.8, 5, 8)
qr12t <- quantileResiduals(VIX, 1, 2, params12t, model="StMAR")

# Non-mixture version of StMAR model
params11t <- c(0.76, 0.93, 1.35, 2.4)
qr11t <- quantileResiduals(VIX, 1, 1, params11t, model="StMAR")
```

```

# G-StMAR model
params12gs <- c(1.5, 0.8, 1.5, 2.9, 0.8, 1.1, 0.6, 3)
qr12gs <- quantileResiduals(VIX, 1, c(1, 1), params12gs, model="G-StMAR")

# Restricted G-StMAR model
params13gsr <- c(1.3, 1, 1.4, 0.8, 0.4, 2, 0.2, 0.25, 0.15, 20)
qr13gsr <- quantileResiduals(VIX, 1, c(2, 1), params13gsr, model="G-StMAR", restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.85, 0.04, 0.3, 3.3, 0.77, 2.8, 0.77)
qr22c <- quantileResiduals(VIX, 2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
qr32trc <- quantileResiduals(VIX, 3, 2, params32trc, model="StMAR",
  restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))

```

quantileResiduals_int *Compute quantile residuals of GMAR, StMAR or G-StMAR model*

Description

quantileResiduals_int computes the quantile residuals of the specified GMAR, StMAR or G-StMAR model.

Usage

```

quantileResiduals_int(data, p, M, params, model = c("GMAR", "StMAR",
  "G-StMAR"), restricted = FALSE, constraints = NULL,
  parametrization = c("intercept", "mean"))

```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
params	a real valued parameter vector specifying the model. For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4)-1 \times 1)$ vector $(\theta, \nu) = (\nu_1, \dots, \nu_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3)+M2-1 \times 1)$ vector $(\theta, \nu) = (\nu_1, \dots, \nu_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M+p-1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M+M2+p-1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.
For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: a size (pxq) constraint matrix **C** of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = \psi_1, \dots, \psi_q$.

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

parametrization is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$?

Value

Returns a $(Tx1)$ numeric vector containing the quantile residuals associated with the specified GMAR, StMAR or G-StMAR model.

Suggested packages

Install the suggested package "gsl" for faster evaluation of the quantile residuals of StMAR and G-StMAR models.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

randomIndividual	Create somewhat random GMAR, StMAR or G-StMAR model compatible parameter vector
------------------	---

Description

randomIndividual creates a somewhat random GMAR, StMAR, G-StMAR model compatible parameter vector.

smartIndividual creates a somewhat random GMAR, StMAR or G-StMAR model compatible parameter vector close to argument params. Sometimes returns exactly the given parameter vector.

Usage

```
randomIndividual(p, M, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, meanscale, sigmascale)
```

```
smartIndividual(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, meanscale, sigmascale, accuracy,
  whichRandom = numeric(0))
```

Arguments

- | | |
|-------|---|
| p | a positive integer specifying the order of AR coefficients. |
| M | For GMAR and StMAR models: a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2. |
| model | is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> . |

restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.</p>
meanscale	a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the $\phi_{m,0}$ or μ_m (depending on the desired parametrization) parameters (for random regimes) should be generated.
sigmascale	a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters (for random regimes) should be generated.
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first $M1$ components are <i>GMAR-type</i> and the rest $M2$ components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
accuracy	a real number larger than zero specifying how close to params the generated parameter vector should be. Standard deviation of the normal distribution from which new parameter values are drawn from will be corresponding parameter value divided by accuracy.

`whichRandom` an (optional) numeric vector of max length M specifying which regimes should be random instead of "smart" when using `smartIndividual`. Does not affect on mixing weight parameters. Default in none.

Details

The functions can be used for example to create initial populations for the genetic algorithm.

Value

Returns estimated parameter vector described in `initpop`.

Examples

```
# GMAR model parameter vector
params22 <- randomIndividual(2, 2, meanscale=c(0, 1), sigmascale=1)
smart22 <- smartIndividual(2, 2, params22, accuracy=10)
cbind(params22, smart22)

# Restricted GMAR parameter vector
params12r <- randomIndividual(1, 2, restricted=TRUE, meanscale=c(-2, 2), sigmascale=2)
smart12r <- smartIndividual(1, 2, params12r, restricted=TRUE, accuracy=20)
cbind(params12r, smart12r)

# StMAR parameter vector: first regime is random in the "smart individual"
params13t <- randomIndividual(1, 3, model="StMAR", meanscale=c(3, 1), sigmascale=3)
smart13t <- smartIndividual(1, 3, params13t, model="StMAR", accuracy=15,
                           meanscale=c(3, 3), sigmascale=3, whichRandom=1)
cbind(params13t, smart13t)

# Restricted StMAR parameter vector
params22tr <- randomIndividual(2, 2, model="StMAR", restricted=TRUE,
                              meanscale=c(3, 2), sigmascale=0.5)
smart22tr <- smartIndividual(2, 2, params22tr, model="StMAR", restricted=TRUE,
                             accuracy=30)
cbind(params22tr, smart22tr)

# G-StMAR parameter vector
params12gs <- randomIndividual(1, c(1, 1), model="G-StMAR", meanscale=c(0, 1),
                              sigmascale=1)
smart12gs <- smartIndividual(1, c(1, 1), params12gs, model="G-StMAR", accuracy=20)
cbind(params12gs, smart12gs)

# Restricted G-StMAR parameter vector
params23gsr <- randomIndividual(2, c(1, 2), model="G-StMAR", restricted=TRUE,
                              meanscale=c(-1, 1), sigmascale=3)
smart23gsr <- smartIndividual(2, c(1, 2), params23gsr, model="G-StMAR", restricted=TRUE,
```

```

                                meanscale=c(0, 1), sigmascale=1, accuracy=20, whichRandom=2)
cbind(params23gsr, smart23gsr)

# GMAR model as a mixture of AR(2) and AR(1) models
C <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- randomIndividual(2, 2, constraints=C, meanscale=c(1, 1),
                              sigmascale=1)
smart22c <- smartIndividual(2, 2, params22c, constraints=C, accuracy=10)
cbind(params22c, smart22c)

# Such constrained StMAR(3, 2) model that the second order AR coefficients
# are constrained to zero.
C0 = matrix(c(1, 0, 0, 0, 0, 1), ncol=2)
C = list(C0, C0)
params32c <- randomIndividual(3, 2, model="StMAR", constraints=C,
                              meanscale=c(1, 1), sigmascale=1)
smart32c <- smartIndividual(3, 2, params32c, model="StMAR", constraints=C, accuracy=10)
cbind(params32c, smart32c)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero. Second regime is random in the "smart individual".
params32trc <- randomIndividual(3, 2, model="StMAR", restricted=TRUE,
                              constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2),
                              meanscale=c(-2, 0.5), sigmascale=4)
smart32trc <- smartIndividual(3, 2, params32trc, model="StMAR", restricted=TRUE,
                              constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2),
                              meanscale=c(0, 0.1), sigmascale=0.1, whichRandom=2,
                              accuracy=20)
cbind(params32trc, smart32trc)

```

randomIndividual_int *Create random GMAR, StMAR or G-StMAR model compatible parameter vector*

Description

randomIndividual_int creates a random GMAR, StMAR or G-StMAR model compatible parameter vector.

smartIndividual_int creates a random GMAR, StMAR or G-StMAR model compatible parameter vector close to argument params.

Usage

```

randomIndividual_int(p, M, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, meanscale, sigmascale)

```

```
smartIndividual_int(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, meanscale, sigmascale, accuracy,
  whichRandom)
```

Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.
- restricted** a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
- constraints** specifies linear constraints applied to the autoregressive parameters.
For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.
For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.
 Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.
- meanscale** a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the $\phi_{m,0}$ or μ_m (depending on the desired parametrization) parameters (for random regimes) should be generated.
- sigmascale** a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters (for random regimes) should be generated.
- params** a real valued parameter vector specifying the model.
For non-restricted models: **For GMAR model:** Size $(M(p+3) - 1x1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.
For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

accuracy	a real number larger than zero specifying how close to params the generated parameter vector should be. Standard deviation of the normal distribution from which new parameter values are drawn from will be corresponding parameter value divided by accuracy.
whichRandom	an (optional) numeric vector of max length M specifying which regimes should be random instead of "smart" when using smartIndividual. Does not affect on mixing weight parameters. Default in none.

Value

Returns estimated parameter vector described in initpop.

random_regime	Create random regime
---------------	----------------------

Description

random_regime generates random regime parameters

Usage

```
random_regime(p, M, meanscale, sigmascale, restricted = FALSE,
              constraints = NULL, m)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2×1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M = M1 + M2$.

meanscale	a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the μ_m mean-parameters are generated in random mutations in the genetic algorithm. Default is <code>c(mean(data), sd(data))</code> . Note that the genetic algorithm optimizes with mean-parametrization even when <code>parametrization="intercept"</code> , but input (in <code>initpop</code>) and output (return value) parameter vectors may be intercept-parametrized.
sigmascale	a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters are generated in the random mutations in the genetic algorithm. Default is <code>var(stats::ar(data, order.max=10)\$resid, na.rm=TRUE)</code> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$. For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = \psi_1, \dots, \psi_q$. Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.
m	which regime?

Value

Regular models: $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$.

Restricted models: Not supported!

Constrained models: Replace the vectors ϕ_m with vectors ψ_m .

reformConstrainedPars *Reform parameter vector with linear constraints to correspond non-constrained parameter vector.*

Description

reformConstrainedPars reforms the parameter vector of a model with linear constraints to the "standard form" so that it's comparable with non-constrained models.

Usage

```
reformConstrainedPars(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	<p>For GMAR and StMAR models: a positive integer specifying the number of mixture components.</p> <p>For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p>For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.</p> <p>For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices C that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.</p> <p>For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.</p> <p>With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case M=1 the parameter α is dropped, and in the case of StMAR or G-StMAR model the degrees of freedom parameters ν_m have to be larger than 2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p>For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.</p> <p>For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.</p> <p>Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is not desired.</p>

Value

Returns such parameter vector corresponding to the input vector that is the form described in params for non-restricted or restricted models (for non-constrained models), and can hence be used just as the parameter vectors of non-constrained models.

reformParameters	<i>Reform any parameter vector into standard form.</i>
------------------	--

Description

reformParameters takes a parameter vector of any (non-constrained) GMAR, StMAR or G-StMAR model and returns a list with the parameter vector in the standard form, parameter matrix containing AR coefficients and component variances, mixing weights alphas and in case of StMAR or G-StMAR model also degrees of freedom parameters.

Usage

```
reformParameters(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE)
```

Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.
For non-restricted models: For GMAR model: Size $(M(p+3) - 1x1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.
For StMAR model: Size $(M(p+4) - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(M(p+3) + M2 - 1x1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
- For restricted models: For GMAR model:** Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.
For StMAR model: Size $(4M + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(3M + M2 + p - 1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

Details

This function does not support models parametrized with general linear constraints! Nor does it have any argument checks.

Value

Returns a list with...

\$params parameter vector in the standard form.

\$pars corresponding parameter matrix containing AR coefficients and component variances. First row for ϕ_0 or means depending on the parametrization. Column for each component.

\$alphas numeric vector containing mixing weights for all components (also for the last one).

\$dfs numeric vector containing degrees of freedom parameters for all components. Returned only if model == "StMAR" or model == "G-StMAR".

reformRestrictedPars *Reform parameter vector with restricted autoregressive parameters to correspond non-restricted parameter vector.*

Description

reformRestrictedPars reforms parameter vector with restricted autoregressive parameters to correspond non-restricted parameter vector.

Usage

```
reformRestrictedPars(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE)
```

Arguments

p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
params	a real valued parameter vector specifying the model.

For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.

For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices \mathbf{C} that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

Value

Returns such parameter vector corresponding to the input vector that is the form described in params for non-restricted models (for non-constrained models). Linear constraints are not supported.

regime_distance	Calculate "distance" between two regimes
-----------------	--

Description

regime_distance scales each regime parameter and then calculates distance between scaled regime_pars1 and regime_pars2.

Usage

```
regime_distance(regime_pars1, regime_pars2)
```

Arguments

- regime_pars1 a numeric vector representing a regime.
 regime_pars2 a numeric vector representing another regime, same length as regime_pars1

Value

Returns "distance" between regime_pars1 and regime_pars2. Values are scaled before calculating the "distance". Read the source code for details.

removeAllConstraints *Transform constrained and restricted parameter vector into the regular form*

Description

removeAllConstraints transforms constrained and restricted parameter vector into the regular form.

Usage

```
removeAllConstraints(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

Arguments

- p a positive integer specifying the order of AR coefficients.
- M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of *GMAR*-type components M1 in the first element and *StMAR*-type components M2 in the second. The total number of mixture components is M=M1+M2.
- params a real valued parameter vector specifying the model.
For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.
For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.
For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.
For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M+M2+p-1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m\psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

Value

Returns such parameter vector corresponding to the input vector that is the form described in params for non-restricted and non-constrained models.

simulateGMAR

simulateGMAR is deprecated

Description

This function is deprecated! Use simulateGSMAR() instead!

Usage

```
simulateGMAR(...)
```

Arguments

... deprecated

simulateGSMAR	<i>Simulate values from GMAR, StMAR or G-StMAR process</i>
---------------	--

Description

simulateGSMAR simulates values from the specified GMAR, StMAR or G-StMAR process.

Usage

```
simulateGSMAR(gsmar, nsimu, initvalues, ntimes = 1)
```

Arguments

gsmar	object of class 'gsmar', generated by function fitGSMAR() or GSMAR().
nsimu	a positive integer specifying how many values will be simulated.
initvalues	a numeric vector with length $\geq p$ specifying the initial values for the simulation. The last element will be used as the initial value for the first lag, the second last element will be initial value for the second lag, etc. If not specified, initial values will be simulated from the process's stationary distribution.
ntimes	a positive integer specifying how many sets of simulations should be performed. If larger than one then only the samples are returned. Uses the same initial values for each set. Default is one.

Value

Returns a list with...

`$sample` a numeric vector containing the simulated values.

`$component` a numeric vector the containing the information from which component each value is generated from.

`$mixingWeights` a size $(nsimu \times M)$ matrix containing the mixing weights corresponding to the sample so that i :th column is for i :th component.

Or if `ntimes > 1` returns a matrix containing the samples so that the i :th sample can be obtained at i :th column.

References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

Examples

```

# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(p=1, M=3, params=params13, model="GMAR")
sim13 <- simulateGSMAR(gmar13, nsimu=500)
ts.plot(sim13$sample)
ts.plot(sim13$component)
ts.plot(sim13$mixingWeights, col=rainbow(3), lty=2)

# Restricted GMAR model
params12r <- c(1.4, 1.8, 0.88, 0.29, 3.18, 0.84)
gmar12r <- GSMAR(p=1, M=2, params=params12r, model="GMAR",
restricted=TRUE)
sim12r <- simulateGSMAR(gmar12r, nsimu=500)
ts.plot(sim12r$sample)
ts.plot(sim12r$component)
ts.plot(sim12r$mixingWeights, col=rainbow(3), lty=2)

# G-StMAR model, with initial values
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(p=1, M=c(1, 1), params=params12gs,
model="G-StMAR")
sim12gs <- simulateGSMAR(gstmar12, nsimu=500, initvalues=5:6)
ts.plot(sim12gs$sample)
ts.plot(sim12gs$component)
ts.plot(sim12gs$mixingWeights, col=rainbow(3), lty=2)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(2.2, 1.8, 0.88, -0.03, 2.4, 0.27, 0.40, 3.9, 1000)
stmar32rc <- GSMAR(data=VIX, p=3, M=2, params=params32trc, model="StMAR",
restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
sim32trc <- simulateGSMAR(stmar32rc, nsimu=500)
ts.plot(sim32trc$sample)
ts.plot(sim32trc$component)
ts.plot(sim32trc$mixingWeights, col=rainbow(3), lty=2)

# Mixture version of Heterogeneous Autoregressive (HAR) model (without data)
paramsHAR <- c(1, 0.1, 0.2, 0.3, 1, 1, 0.15, 0.25, 0.35, 1, 0.55)
r1 = c(1, rep(0, 21)); r2 = c(rep(0.2, 5), rep(0, 17)); r3 = rep(1/22, 22)
R0 = cbind(r1, r2, r3)
mixhar <- GSMAR(p=22, M=2, params=paramsHAR, model="GMAR", constraints=list(R0, R0))
simhar <- simulateGSMAR(mixhar, nsimu=1000)
ts.plot(simhar$sample)
ts.plot(simhar$component)
ts.plot(simhar$mixingWeights, col=rainbow(3), lty=2)

```

sortComponents	<i>Sort the mixture components of GMAR, StMAR or G-StMAR model</i>
----------------	--

Description

sortComponents sorts mixture components of the specified GMAR, StMAR or G-StMAR model by the mixing weights when the parameter vector is in the "standard form" for restricted or non-restricted models.

Usage

```
sortComponents(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE)
```

Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.
For G-StMAR model: a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.
For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$.
For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices **C** that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.
For restricted models: For GMAR model: Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.
For StMAR model: Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.
For G-StMAR model: Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.
With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix **C** that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

Details

This function does not support models parametrized with general linear constraints!
Models with general linear constraints are not supported.

Value

Returns a parameter vector sorted by it's mixing weights, described in params.

standardErrors	<i>Calculate standard errors for estimates of GMAR, StMAR or GStMAR model</i>
----------------	---

Description

standardErrors numerically approximates standard errors for the given estimates of GMAR, StMAR or GStMAR model

Usage

```
standardErrors(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), minval)
```

Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	For GMAR and StMAR models: a positive integer specifying the number of mixture components. For G-StMAR model: a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
params	a real valued parameter vector specifying the model. For non-restricted models: For GMAR model: Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$, where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$, $m = 1, \dots, M$. For StMAR model: Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(M(p+3)+M2-1x1)$ vector $(\theta, \nu) = (\nu_1, \dots, \nu_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$.

With linear constraints: Replace the vectors ϕ_m with vectors ψ_m and provide a list of constraint matrices \mathbf{C} that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$, where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: For GMAR model: Size $(3M + p - 1x1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$, where $\phi = (\phi_1, \dots, \phi_M)$.

For StMAR model: Size $(4M+p-1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

For G-StMAR model: Size $(3M+M2+p-1x1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$.

With linear constraints: Replace the vector ϕ with vector ψ and provide a constraint matrix C that satisfies $\phi = R\psi$, where $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient, σ^2 a variance, α a mixing weight and ν a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter α is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters ν_m have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

For non-restricted models: a list of size (pxq_m) constraint matrices C_m of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$, where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$.

For restricted models: a size (pxq) constraint matrix C of full column rank satisfying $\phi = C\psi$, where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$.

Symbol ϕ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

conditional a logical argument specifying whether the conditional or exact log-likelihood function should be used.

parametrization is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$?

minval this will be returned when the parameter vector is outside the parameter space.

Value

Approximate standard errors of the parameter values

swap_parametrization *Swap the parametrization of object of class 'gsmar' defining a gsmar model*

Description

swap_parametrization swaps the parametrization of object of class 'gsmar' to "mean" if the current parametrization is "intercept", and vice versa.

Usage

```
swap_parametrization(gsmar, calc_std_errors = TRUE)
```

Arguments

gsmar object of class 'gsmar', generated by function fitGSMAR() or GSMAR().
 calc_std_errors should approximate standard errors be calculated?

Details

swap_parametrization() is convenient tool if you have estimated the model in "intercept"-parametrization, but wish to work with "mean"-parametrization in the future, or vice versa. In gsmarkit, for example the approximate standard errors are only available for parametrized parameters.

References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

See Also

[fitGSMAR](#), [GSMAR](#), [iterate_more](#), [get_gradient](#), [get_regime_means](#)

Examples

```
# GMAR model with intercept parametrization
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(data=VIX, p=1, M=3, params=params13, model="GMAR")
gmar13

# Swap to mean parametrization
gmar13 <- swap_parametrization(gmar13)
gmar13
```

VIX

CBOE Volatility Index: VIX

Description

A dataset containing daily CBOE Volatility Index from 5th of July 2016 to 4th of July 2017.

Usage

VIX

Format

A numeric vector containing 252 values.

Source

<https://fred.stlouisfed.org/series/VIXCLS>

Index

*Topic **datasets**

- VIX, 89

- aa_uGMAR, 3
- aa_uGMAR-package (aa_uGMAR), 3
- add_data, 3, 23, 35
- add_dfs, 4
- all_pos_ints, 5

- calc_gradient, 6, 43
- calc_hessian (calc_gradient), 6
- change_parametrization, 9
- changeRegime, 7
- check_data, 13
- check_gsmar, 13
- check_model, 14
- check_params_length, 14
- checkAndCorrectData, 11
- checkConstraintMat, 11
- checkPM, 12

- diagnosticPlot, 15, 23, 61, 63

- extractRegime, 18

- fitGMAR, 20
- fitGSMAR, 4, 17, 20, 35, 43, 61, 63, 88
- forecastGMAR, 24
- format_valuef, 25

- GAFit, 25
- get_ar_roots, 30
- get_gradient, 4, 23, 35, 88
- get_gradient (calc_gradient), 6
- get_hessian (calc_gradient), 6
- get_IC, 31
- get_regime_means, 4, 31, 88
- getOmega, 28
- GSMAR, 4, 17, 23, 32, 43, 61, 63, 88

- isIdentifiable (isStationary_int), 38

- isStationary, 36
- isStationary_int, 38
- iterate_more, 4, 23, 35, 40, 88

- logLik.gsmar (GSMAR), 32
- loglikelihood, 41
- loglikelihood_int, 44

- mixingWeights, 43, 47
- mixingWeights_int, 49

- nParams, 51

- parameterChecks, 52
- pick_alphas, 53
- pick_dfs, 55
- pick_pars, 56
- pick_phi0, 57
- plot.gsmar (GSMAR), 32
- plot.gsmarpred, 59
- plot.qrtest, 59
- plotGMAR, 62
- predict.gsmar, 17, 23, 35, 61, 62
- print.gsmar (GSMAR), 32
- print.gsmarpred, 64
- print.gsmarsum, 65
- print.qrtest (plot.qrtest), 59

- quantileResiduals, 43, 65
- quantileResiduals_int, 68
- quantileResidualTests, 17, 23, 63
- quantileResidualTests (plot.qrtest), 59

- random_regime, 75
- randomIndividual, 70
- randomIndividual_int, 73
- reformConstrainedPars, 76
- reformParameters, 78
- reformRestrictedPars, 79
- regime_distance, 80
- removeAllConstraints, 81

residuals.gsmar (GSMAR), [32](#)

simulateGMAR, [82](#)

simulateGSMAR, [23](#), [35](#), [83](#)

smartIndividual (randomIndividual), [70](#)

smartIndividual_int
 (randomIndividual_int), [73](#)

sortComponents, [85](#)

standardErrors, [86](#)

summary.gsmar (GSMAR), [32](#)

swap_parametrization, [23](#), [35](#), [88](#)

VIX, [89](#)