

# Package ‘gmvarkit’

November 3, 2020

**Title** Estimate Gaussian Mixture Vector Autoregressive Model

**Version** 1.2.3

**Description**

Unconstrained and constrained maximum likelihood estimation of structural and reduced form Gaussian mixture vector autoregressive (GMVAR) model, quantile residual tests, graphical diagnostics, simulations, forecasting, and estimation of generalized impulse response function. Leena Kalliovirta, Mika Meitz, Pentti Saikkonen (2016) <doi:10.1016/j.jeconom.2016.02.012>, Savi Virolainen (2020) <arXiv:2007.04713>.

**Depends** R (>= 4.0.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** Brodningnag (>= 1.2-6), mvnfast (>= 0.2.5), parallel (>= 4.0.0), stats (>= 4.0.0), pbapply (>= 1.4-2), graphics (>= 4.0.0), grDevices (>= 4.0.0)

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Savi Virolainen [aut, cre]

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

**Repository** CRAN

**Date/Publication** 2020-11-03 10:30:02 UTC

## R topics documented:

add_data . . . . .	3
all_pos_ints . . . . .	5
alt_gmvar . . . . .	6
calc_gradient . . . . .	7

change_parametrization . . . . .	9
change_regime . . . . .	11
check_constraints . . . . .	13
check_data . . . . .	14
check_gmvar . . . . .	15
check_null_data . . . . .	15
check_parameters . . . . .	16
check_pMd . . . . .	18
cond_moments . . . . .	19
cond_moment_plot . . . . .	21
diagnostic_plot . . . . .	23
diag_Omegas . . . . .	24
dlogmultinorm . . . . .	25
eurusd . . . . .	26
fitGMVAR . . . . .	27
format_valuef . . . . .	32
form_boldA . . . . .	32
GAfit . . . . .	33
get_boldA_eigens . . . . .	38
get_IC . . . . .	39
get_minval . . . . .	39
get_omega_eigens . . . . .	40
get_regime_autocovs . . . . .	41
get_regime_autocovs_int . . . . .	42
get_regime_means . . . . .	44
get_regime_means_int . . . . .	45
get_test_Omega . . . . .	47
get_unconstrained_structural_pars . . . . .	50
GIRF . . . . .	51
GMVAR . . . . .	54
gmvarkit . . . . .	58
gmvar_to_sgmvar . . . . .	59
in_paramspace . . . . .	60
in_paramspace_int . . . . .	62
is_stationary . . . . .	64
iterate_more . . . . .	66
loglikelihood . . . . .	68
loglikelihood_int . . . . .	71
LR_test . . . . .	74
n_params . . . . .	75
pick_allA . . . . .	76
pick_all_phi0_A . . . . .	78
pick_alphas . . . . .	80
pick_Am . . . . .	81
pick_Ami . . . . .	83
pick_lambdas . . . . .	85
pick_Omegas . . . . .	86
pick_phi0 . . . . .	88

pick_regime . . . . .	89
pick_W . . . . .	91
plot.gmvarpred . . . . .	93
plot.qrtest . . . . .	94
predict.gmvar . . . . .	96
print.gmvarpred . . . . .	98
print.gmvarsum . . . . .	99
print_std_errors . . . . .	99
profile_logliks . . . . .	101
quantile_residuals . . . . .	103
quantile_residuals_int . . . . .	105
random_coefmats . . . . .	107
random_coefmats2 . . . . .	108
random_covmat . . . . .	109
random_ind . . . . .	110
random_ind2 . . . . .	113
reform_constrained_pars . . . . .	116
reform_data . . . . .	118
reform_structural_pars . . . . .	118
regime_distance . . . . .	120
reorder_W_columns . . . . .	121
simulateGMVAR . . . . .	122
smart_covmat . . . . .	125
smart_ind . . . . .	126
sort_components . . . . .	129
standard_errors . . . . .	131
swap_parametrization . . . . .	134
swap_W_signs . . . . .	135
uncond_moments . . . . .	137
uncond_moments_int . . . . .	138
unvec . . . . .	140
unvech . . . . .	141
unWvec . . . . .	142
vec . . . . .	143
vech . . . . .	143
Wald_test . . . . .	144
Wvec . . . . .	145

**Index****147**


---

add_data	<i>Add data to an object of class 'gmvar' defining a GMVAR model</i>
----------	--

---

**Description**

add\_data adds or updates data to object of class 'gmvar' that defines a GMVAR model. Also calculates mixing weights and quantile residuals accordingly.

**Usage**

```
add_data(data, gmvar, calc_cond_moments = TRUE, calc_std_errors = FALSE)
```

**Arguments**

**data** a matrix or class 'ts' object with  $d > 1$  columns. Each column is taken to represent a single time series. NA values are not supported.

**gmvar** an object of class 'gmvar' created with `fitGMVAR` or `GMVAR`.

**calc\_cond\_moments** should conditional means and covariance matrices should be calculated? Default is TRUE if the model contains data and FALSE otherwise.

**calc\_std\_errors** should approximate standard errors be calculated?

**Value**

Returns an object of class 'gmvar' defining the specified GMVAR 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. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [iterate\\_more](#)

**Examples**

```
# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(p=1, M=2, d=2, params=params122)
mod122

mod122_2 <- add_data(data, mod122)
mod122_2

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
```

```
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.031, 2.356, 1.786, 3.000, 1.250, 0.060, 0.036,
  1.335, -0.290, -0.083, -0.047, -0.356, 0.934, -0.152, 5.201, 5.883,
  3.560, 9.799, 0.368)
mod222c <- GMVAR(p=2, M=2, d=2, params=params222c, constraints=C_mat)
mod222c

mod222c_2 <- add_data(data, mod222c)
mod222c_2

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
  -0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
  0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(p=2, M=2, d=2, params=params222s, structural_pars=list(W=W_222))
mod222s

mod222s_2 <- add_data(data, mod222s)
mod222s_2
```

---

all\_pos\_ints

*Check whether all arguments are positive integers*

---

## Description

all\_pos\_ints checks whether all the elements in a vector are positive integers.

## Usage

```
all_pos_ints(x)
```

## Arguments

x                    a vector containing the elements to be tested.

## Value

Returns TRUE or FALSE accordingly.

---

alt_gmvar	<i>Construct a GMVAR model based on results from an arbitrary estimation round of fitGMVAR</i>
-----------	--

---

### Description

alt\_gmvar constructs a GMVAR model based on results from an arbitrary estimation round of fitGMVAR.

### Usage

```
alt_gmvar(
  gmvar,
  which_round = 1,
  which_largest,
  calc_cond_moments = TRUE,
  calc_std_errors = TRUE
)
```

### Arguments

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
which_round	based on which estimation round should the model be constructed? An integer value in 1,...,ncalls.
which_largest	based on estimation round with which largest log-likelihood should the model be constructed? An integer value in 1,...,ncalls. For example, which_largest=2 would take the second largest log-likelihood and construct the model based on the corresponding estimates. If used, then which_round is ignored.
calc_cond_moments	should conditional means and covariance matrices should be calculated? Default is TRUE if the model contains data and FALSE otherwise.
calc_std_errors	should approximate standard errors be calculated?

### Details

It's sometimes useful to examine other estimates than the one with the highest log-likelihood. This function is wrapper around GMVAR that picks the correct estimates from an object returned by fitGMVAR.

### Value

Returns an object of class 'gmvar' defining the specified reduced form or structural GMVAR model. Can be used to work with other functions provided in gmvarkit.

Remark that the first autocovariance/correlation matrix in \$uncond\_moments is for the lag zero, the second one for the lag one, etc.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

## See Also

[fitGMVAR](#), [GMVAR](#), [iterate\\_more](#)

## Examples

```
# These are long running examples and use parallel computing
data(eurusd, package="gmvarkit")
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2) model
fit12 <- fitGMVAR(data, p=1, M=2, ncalls=2, seeds=7:8)
fit12
fit12_2 <- alt_gmvar(fit12, which_round=1)
fit12_2

# Structural GMVAR(1,2) model identified with sign
# constraints.
W_122 <- matrix(c(1, 1, -1, 1), nrow=2)
fit12s <- fitGMVAR(data, p=1, M=2, structural_pars=list(W=W_122),
  ncalls=2, seeds=c(1, 42))
fit12s
fit12s_2 <- alt_gmvar(fit12s, which_largest=2)
fit12s_2
```

---

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 a class 'gmvar' object. get\_soc returns eigenvalues of the Hessian matrix, and get\_foc is the same as get\_gradient but named conveniently.

**Usage**

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

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

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

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

```
get_soc(gmvar, h = 6e-06)
```

```
get_foc(gmvar, h = 6e-06)
```

**Arguments**

x	a numeric vector specifying the point where the gradient or Hessian should be calculated.
fn	a function that takes in argument x as the <b>first</b> argument.
h	difference used to approximate the derivatives.
...	other arguments passed to fn
gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.

**Details**

In particular, the functions `get_foc` and `get_soc` can be used to check whether the found estimates denote a (local) maximum point, a saddle point, or something else. Note that profile log-likelihood functions can be conveniently plotted with the function `profile_logliks`.

**Value**

Gradient functions return numerical approximation of the gradient and Hessian functions return numerical approximation of the Hessian. `get_soc` returns eigenvalues of the Hessian matrix.

**Warning**

No argument checks!

**See Also**

[profile\\_logliks](#)

**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)

# These examples below use the data 'eurusd' which comes
# with the package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(data, p=1, M=2, params=params122)
get_gradient(mod122)
get_hessian(mod122)
get_soc(mod122)

```

---

change\_parametrization

*Change parametrization of a parameter vector*

---

## Description

change\_parametrization changes the parametrization of the given parameter vector to change\_to.

## Usage

```

change_parametrization(
  p,
  M,
  d,
  params,
  constraints = NULL,
  structural_pars = NULL,
  change_to = c("intercept", "mean")
)

```

## Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1)x1)$  and have form  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1)x1)$

and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2xq)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

constraints

a size  $(Mpd^2xq)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))(pd^2x1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi (qx1)$  contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$   $(Mpd^2xpd^2)$  where  $I = \text{diag}(p*d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be applied.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

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. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

change_regime	<i>Change regime parameters <math>v_m = (\phi_m, 0, \phi_m, \sigma_m)</math> of the given parameter vector</i>
---------------	--

---

### Description

change\_regime changes the regime parameters (excluding mixing weights parameter) of the pointed regime to the new given parameters.

### Usage

```
change_regime(p, M, d, params, m, regime_pars, structural_pars = NULL)
```

### Arguments

p a positive integer specifying the autoregressive order of the model.  
M a positive integer specifying the number of mixture components.  
d number of time series in the system, i.e. the dimension.  
params a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1)x1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2qxq)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M - 1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

m which component?

regime\_pars **For reduced form models:** a size  $((pd^2 + d + d(d + 1)/2)x1)$  vector  $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$ .

**For structural models:** a length  $pd^2 + d$  vector  $(\phi_{m,0}, \phi_m)$ .

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C_\lambda$  - a  $(d(M - 1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C_\lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Details

Does not currently support models with AR or lambda parameter constraints.

### Value

Returns parameter vector with  $m$ :th regime changed to regime\_pars.

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

check_constraints	<i>Check the constraint matrix has the correct form</i>
-------------------	---

---

### Description

check\_constraints checks that the constraints are correctly set.

### Usage

```
check_constraints(p, M, d, constraints = NULL, structural_pars = NULL)
```

### Arguments

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.
constraints	a size $(Mpd^2 \times q)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2 \times 1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi$ ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ ( $Mpd^2 \times pd^2$ ) where $I = diag(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.

**structural\_pars**

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Details**

If `is.null(constraints)`, then this function doesn't do anything.

**Value**

Checks the constraint matrix **C** and throws an error if something is wrong.

---

check\_data

*Check the data is in the correct form*

---

**Description**

check\_data checks the data.

**Usage**

```
check_data(data, p)
```

**Arguments**

**data** a matrix or class 'ts' object with  $d > 1$  columns. Each column is taken to represent a single time series. NA values are not supported.

**p** a positive integer specifying the autoregressive order of the model.

**Value**

Checks the data and tries to correct it. Throws an error if something is wrong and returns the corrected data otherwise.

---

check_gmvar	<i>Checks whether the given object has class attribute 'gmvar'</i>
-------------	--

---

**Description**

check\_gmvar checks that the object has class attribute 'gmvar'.

**Usage**

```
check_gmvar(object, object_name)
```

**Arguments**

object	S3 object to be tested
object_name	what is the name of the object that should of class 'gmvar'?

**Value**

Throws an error if the object doesn't have the class attribute 'gmvar'.

---

check_null_data	<i>Checks whether the given object contains data</i>
-----------------	--

---

**Description**

check\_null\_data checks that the gmvar object has data.

**Usage**

```
check_null_data(gmvar)
```

**Arguments**

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
-------	--

**Value**

Throws an error if is.null(gmvar\$data).

---

check_parameters	<i>Check that the given parameter vector satisfies the model assumptions</i>
------------------	--

---

### Description

check\_parameters checks whether the given parameter vector satisfies the model assumptions. Does NOT consider the identifiability condition!

### Usage

```
check_parameters(p, M, d, params, constraints = NULL, structural_pars = NULL)
```

### Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1)x1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1)x1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2xq)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M - 1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .



$vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by *Kalliovirta, Meitz and Saikkonen (2016)* introducing the GMVAR model.

**constraints** a size  $(Mpd^2xq)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi$  ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2xpd^2$ ) where  $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

**structural\_pars**

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

## Value

Throws an informative error if there is something wrong with the parameter vector.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

## Examples

```
## Not run:
# These examples will cause an informative error

# GMVAR(1, 1), d=2 model:
params112 <- c(1.07, 127.71, 0.99, 0.00, -0.01, 1.00, 4.05,
              2.22, 8.87)
check_parameters(p=1, M=1, d=2, params=params11)
```

```

# GMVAR(2, 2), d=2 model:
params222 <- c(1.39, -0.77, 1.31, 0.14, 0.09, 1.29, -0.39,
  -0.07, -0.11, -0.28, 0.92, -0.03, 4.84, 1.01, 5.93, 1.25,
  0.08, -0.04, 1.27, -0.27, -0.07, 0.03, -0.31, 5.85, 10.57,
  9.84, 0.74)
check_parameters(p=2, M=2, d=2, params=params222)

# GMVAR(2, 2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.03, 2.36, 1.79, 3.00, 1.25, 0.06,0.04,
  1.34, -0.29, -0.08, -0.05, -0.36, 0.93, -0.15, 5.20,
  5.88, 3.56, 9.80, 1.37)
check_parameters(p=2, M=2, d=2, params=params222c, constraints=C_mat)

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
  -0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
  0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
check_parameters(p=2, M=2, d=2, params=params222s,
  structural_pars=list(W=W_222))

## End(Not run)

```

---

check\_pMd

*Check that p, M, and d are correctly set*

---

## Description

check\_pMd checks the arguments p, M, and d.

## Usage

```
check_pMd(p, M, d)
```

## Arguments

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.

## Value

Throws an error if something is wrong.

---

cond_moments	<i>Compute conditional moments of a GMVAR model</i>
--------------	---

---

### Description

loglikelihood compute conditional regimewise means, conditional means, and conditional covariance matrices of a GMVAR model.

### Usage

```
cond_moments(
  data,
  p,
  M,
  params,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL,
  to_return = c("regime_cmeans", "total_cmeans", "total_ccovs")
)
```

### Arguments

**data** a matrix or class 'ts' object with  $d > 1$  columns. Each column is taken to represent a single time series. NA values are not supported.

**p** a positive integer specifying the autoregressive order of the model.

**M** a positive integer specifying the number of mixture components.

**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If W is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

parametrization

"mean" or "intercept" determining whether the model is parametrized with regime means  $\mu_m$  or intercept parameters  $\phi_{m,0}$ ,  $m=1, \dots, M$ .

constraints

a size  $(Mpd^2xq)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi$  ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2xpd^2$ ) where  $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new ( $rx1$ ) parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

to\_return

should the regimewise conditional means, total conditional means, or total conditional covariance matrices be returned?

## Details

The first  $p$  values are used as the initial values, and by conditional we mean conditioning on the past. Formulas for the conditional means and covariance matrices are given in equations (3) and (4) of KMS (2016).

**Value**

**If** to\_return=="regime\_cmeans": an [T-p, d, M] array containing the regimewise conditional means (the first p values are used as the initial values).

**If** to\_return=="total\_cmeans": a [T-p, d] matrix containing the conditional means of the process (the first p values are used as the initial values).

**If** to\_return=="total\_ccov": an [d, d, T-p] array containing the conditional covariance matrices of the process (the first p values are used as the initial values).

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

Other moment functions: [get\\_regime\\_autocovs\(\)](#), [get\\_regime\\_means\(\)](#), [uncond\\_moments\(\)](#)

**Examples**

```
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
-0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
9.838, 0.740)
cond_moments(data=data, p=2, M=2, params=params222, parametrization="mean",
to_return="regime_cmeans")
cond_moments(data=data, p=2, M=2, params=params222, parametrization="mean",
to_return="total_cmeans")
cond_moments(data=data, p=2, M=2, params=params222, parametrization="mean",
to_return="total_ccovs")
```

---

cond\_moment\_plot

*Conditional mean or variance plot for a GMVAR model*


---

**Description**

cond\_moment\_plot plots the one-step in-sample conditional means/variances of the model along with the individual time series contained in the model (e.g. the time series the model was fitted to). Also plots the regimewise conditional means/variances multiplied with mixing weights.

**Usage**

```
cond_moment_plot(
  gmvar,
  which_moment = c("mean", "variance"),
  grid = FALSE,
  ...
)
```

**Arguments**

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
which_moment	should conditional means or variances be plotted?
grid	add grid to the plots?
...	additional paramters passed to grid(...) plotting the grid if grid == TRUE.

**Details**

The conditional mean plot works best if the data contains positive values only. acf from the package stats and the plot method for class 'acf' objects is employed.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[profile\\_logliks](#), [fitGMVAR](#), [GMVAR](#), [quantile\\_residual\\_tests](#), [LR\\_test](#), [Wald\\_test](#), [diagnostic\\_plot](#)

**Examples**

```
# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(2,2), d=2 model:
params222 <- c(1.386, -0.765, 1.314, 0.145, 0.094, 1.292, -0.389,
  -0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 1.005, 5.928, 1.248,
  0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
  9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222)

cond_moment_plot(mod222, which_moment="mean")
cond_moment_plot(mod222, which_moment="variance")
cond_moment_plot(mod222, which_moment="mean", grid=TRUE, lty=2)
```

---

diagnostic_plot	<i>Quantile residual diagnostic plot for a GMVAR model</i>
-----------------	--

---

### Description

diagnostic\_plot plots a multivariate quantile residual diagnostic plot for either autocorrelation, conditional heteroskedasticity, or normality, or simply draws the quantile residual time series.

### Usage

```
diagnostic_plot(
  gmvar,
  type = c("all", "series", "ac", "ch", "norm"),
  maxlag = 10,
  wait_time = 4
)
```

### Arguments

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
type	which type of diagnostic plot should be plotted? <ul style="list-style-type: none"> <li>• "all" all below sequentially.</li> <li>• "series" the quantile residual time series.</li> <li>• "ac" the quantile residual autocorrelation and cross-correlation functions.</li> <li>• "ch" the squared quantile residual autocorrelation and cross-correlation functions.</li> <li>• "norm" the quantile residual histogram with theoretical standard normal density (dashed line) and standard normal QQ-plots.</li> </ul>
maxlag	the maximum lag considered in types "ac" and "ch".
wait_time	if type == all how many seconds to wait before showing next figure?

### Details

Auto- and cross-correlations (types "ac" and "ch") are calculated with the function acf from the package stats and the plot method for class 'acf' objects is employed.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[profile\\_logliks](#), [fitGMVAR](#), [GMVAR](#), [quantile\\_residual\\_tests](#), [LR\\_test](#), [Wald\\_test](#), [cond\\_moment\\_plot](#), [acf](#), [density](#), [predict.gmvar](#)

**Examples**

```
# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(data, p=1, M=2, params=params122)
diagnostic_plot(mod122, type="series")
diagnostic_plot(mod122, type="ac")

# GMVAR(2,2), d=2 model:
params222 <- c(1.386, -0.765, 1.314, 0.145, 0.094, 1.292, -0.389,
  -0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 1.005, 5.928, 1.248,
  0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
  9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222)
diagnostic_plot(mod222, type="ch")
diagnostic_plot(mod222, type="norm")

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.031, 2.356, 1.786, 3.000, 1.250, 0.060, 0.036,
  1.335, -0.290, -0.083, -0.047, -0.356, 0.934, -0.152, 5.201, 5.883,
  3.560, 9.799, 0.368)
mod222c <- GMVAR(data, p=2, M=2, params=params222c, constraints=C_mat)
diagnostic_plot(mod222c, wait_time=0.1)
diagnostic_plot(mod222c, type="ac", maxlag=12)
```

---

diag\_Omegas

*Simultaneously diagonalize two covariance matrices*


---

**Description**

diag\_Omegas Simultaneously diagonalizes two covariance matrices using eigenvalue decomposition.

**Usage**

```
diag_Omegas(Omega1, Omega2)
```



**Arguments**

Omega1            a positive definite ( $d \times d$ ) covariance matrix ( $d > 1$ )  
 Omega2            another positive definite ( $d \times d$ ) covariance matrix

**Details**

See the return value and Muirhead (1982), Theorem A9.9 for details.

**Value**

Returns a length  $d^2 + d$  vector where the first  $d^2$  elements are  $vec(W)$  with the columns of  $W$  being (specific) eigenvectors of the matrix  $\Omega_2 \Omega_1^{-1}$  and the rest  $d$  elements are the corresponding eigenvalues "lambdas". The result satisfies  $WW' = Omega1$  and  $Wdiag(lambdas)W' = Omega2$ .

If Omega2 is not supplied, returns a vectorized symmetric (and pos. def.) square root matrix of Omega1.

**Warning**

No argument checks! Does not work with dimension  $d = 1$ !

**References**

- Muirhead R.J. 1982. Aspects of Multivariate Statistical Theory, Wiley.

**Examples**

```
d <- 2
W0 <- matrix(1:(d^2), nrow=2)
lambdas0 <- 1:d
(Omg1 <- W0%*%t(W0))
(Omg2 <- W0%*%diag(lambdas0)%*%t(W0))
res <- diag_Omegas(Omg1, Omg2)
W <- matrix(res[1:(d^2)], nrow=d, byrow=FALSE)
tcrossprod(W) # == Omg1
lambdas <- res[(d^2 + 1):(d^2 + d)]
W%*%diag(lambdas)%*%t(W) # == Omg2
```

---

dlogmultinorm

*Calculate logarithms of multiple multivariate normal densities with varying mean and constant covariance matrix*

---

**Description**

dlogmultinorm calculates logarithms of multiple multivariate normal densities with varying mean and constant covariance matrix.

**Usage**

```
dlogmultinorm(y, mu, Omega)
```

**Arguments**

**y** dimension  $(Txk)$  matrix where each row is a  $k$ -dimensional random vector

**mu** dimension  $(Txk)$  matrix where each row is the mean of the  $k$ -dimensional random vector in corresponding row of **y**.

**Omega** the  $(k \times k)$  covariance matrix Omega.

**Value**

Returns a size  $(Tx1)$  vector containing the multinormal densities in logarithm.

---

eurusd	<i>Euro area and U.S. long-term government bond yields and Euro-U.S. dollar exchange rate.</i>
--------	--

---

**Description**

A dataset containing time series of the difference between the monthly Euro area and U.S. long-term government bond yields and monthly average Euro - U.S. dollar exchange rate. The data covers the time period January 1989 - December 2009 with monthly frequency. This is the same data (in non-scaled form) that is used by Kalliovirta et. al. (2016).

**Usage**

```
eurusd
```

**Format**

A numeric matrix of class 'ts' with 252 rows and 2 columns with one time series in each column:

**First column:** The difference between the monthly Euro area and U.S. long-term government bond yields (10 year maturity,  $i_{euro} - i_{us}$ ), from January 1989 to December 2009. calculated by the ECB and the Federal Reserve Board; prior to 2001, the Euro area data refer to the "EU11" countries, and afterwards with changing composition eventually to the "EU17" by the end of the data period.

**Second column:** Monthly average Euro - U.S. dollar exchange rate, from January 1989 to December 2009. Based on the ECU - USD exchange rate prior to 1999.

**Source**

OECD Statistics

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

 fitGMVAR

*Two-phase maximum likelihood estimation of a GMVAR model*


---

## Description

fitGMVAR estimates a GMVAR model in two phases: in the first phase it uses a genetic algorithm to find starting values for a gradient based variable metric algorithm, which it then uses to finalize the estimation in the second phase. Parallel computing is utilized to perform multiple rounds of estimations in parallel.

## Usage

```
fitGMVAR(
  data,
  p,
  M,
  conditional = TRUE,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL,
  ncalls = round(10 + 9 * log(M)),
  ncores = min(2, ncalls, parallel::detectCores()),
  maxit = 300,
  seeds = NULL,
  print_res = TRUE,
  ...
)
```

## Arguments

data	a matrix or class 'ts' object with $d > 1$ columns. Each column is taken to represent a single time series. NA values are not supported.
p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
conditional	a logical argument specifying whether the conditional or exact log-likelihood function

parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}$ , $m=1,\dots,M$ .
constraints	a size $(Mpd^2xq)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi$ ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ ( $Mpd^2xpd^2$ ) where $I = \text{diag}(p*d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	If NULL a reduced form model is considered. For structural model, should be a list containing the following elements: <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(dxd)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• <math>C\_lambda</math> - a <math>(d(M-1)xr)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new <math>(rx1)</math> parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of <math>C\_lambda</math> must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul> <p>See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is <math>W</math> times a time-varying diagonal matrix with positive diagonal entries).</p>
ncalls	the number of estimation rounds that should be performed.
ncores	the number CPU cores to be used in parallel computing.
maxit	the maximum number of iterations in the variable metric algorithm.
seeds	a length <code>ncalls</code> vector containing the random number generator seed for each call to the genetic algorithm, or NULL for not initializing the seed. Exists for creating reproducible results.
print_res	should summaries of estimation results be printed?
...	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 certain** 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 or saddle point instead. Therefore, a (sometimes large) number of estimation rounds is required for reliable results. Because of the nature of the model, the estimation may fail especially in the cases where the number of mixture components is chosen too large.

The estimation process is computationally heavy and it might take considerably long time for large models with large number of observations. If the iteration limit `maxit` in the variable metric algorithm is reached, one can continue the estimation by iterating more with the function `iterate_more`.

Alternatively, one may use the found estimates as starting values for the genetic algorithm and employ another round of estimation (see ?GAfit how to set up an initial population with the dot parameters).

**If the estimation algorithm fails to create an initial population for the genetic algorithm**, it usually helps to scale the individual series so that the AR coefficients (of a VAR model) will be relative small, preferably less than one. Even if one is able to create an initial population, it should be preferred to scale the series so that most of the AR coefficients will not be very large, as the estimation algorithm works better with small AR coefficients. If needed, another package can be used to fit linear VARs to the series to see which scaling of the series results in relatively small AR coefficients.

The code of the genetic algorithm is mostly based on the description by *Dorsey and Mayer (1995)* but it includes some extra features that were found useful for this particular estimation problem. For instance, the genetic algorithm uses a slightly modified version of the 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 gradient based variable metric algorithm used in the second phase is implemented with function `optim` from the package `stats`.

Finally, note that the structural models are even more difficult to estimate than the reduced form models due to the different parametrization of the covariance matrices. If necessary, an initial population may be constructed for the genetic algorithm based on the estimation results of a reduced form model. It does not seem unambiguous, however, how to do that so that the (structural) parameter constraints are satisfied. Also, be aware that if the lambda parameters are constrained in any other way than by restricting some of them to be identical, the parameter "lambda\_scale" of the genetic algorithm (see ?GAfit) needs to be carefully adjusted accordingly. Be aware that if a structural model is considered and the lambda parameters are constrained in some other way than constraining some of them to be identical, the settings of the genetic algorithm may have to be adjusted.

## Value

Returns an object of class 'gmvar' defining the estimated (reduced form or structural) GMVAR model. Multivariate quantile residuals (Kalliovirta and Saikkonen 2010) are also computed and included in the returned object. In addition, the returned object contains the estimates and log-likelihood values from all the estimation rounds performed. The estimated parameter vector can be obtained at `gmvar$params` (and corresponding approximate standard errors at `gmvar$std_errors`) and it is...

**For unconstrained models:** ...a size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1)x1)$  vector that has form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** ...a size  $((M(d + d(d + 1)/2 + 1) + q - 1)x1)$  vector that has form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2xq)$  constraint matrix.

**For structural GMVAR model:** ...a vector that has the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M, \alpha_1, \dots, \alpha_{M-1})$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi$  ( $qx1$ ) that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

Remark that the first autocovariance/correlation matrix in \$uncond\_moments is for the lag zero, the second one for the lag one, etc.

### S3 methods

The following S3 methods are supported for class 'gmvar': logLik, residuals, print, summary, predict and plot.

### 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. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- 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.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

### See Also

GMVAR, iterate\_more, predict.gmvar, profile\_logliks, simulateGMVAR, quantile\_residual\_tests, print\_std\_errors, swap\_parametrization, get\_gradient, GIRF, LR\_test, Wald\_test, gmvar\_to\_sgmvar, reorder\_W\_columns, swap\_W\_signs, cond\_moment\_plot

**Examples**

```

## These are long running examples that use parallel computing!

# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form (similar to Kalliovirta et al. 2016).
data(eurusd, package="gmvarkit")
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2) model: 10 estimation rounds with seeds set
# for reproducibility
fit12 <- fitGMVAR(data, p=1, M=2, ncalls=10, seeds=1:10)
fit12
plot(fit12)
summary(fit12)
print_std_errors(fit12)
profile_logliks(fit12)

# Structural GMVAR(1,2) model identified with sign
# constraints.
W_122 <- matrix(c(1, 1, -1, 1), nrow=2)
fit12s <- fitGMVAR(data, p=1, M=2, structural_pars=list(W=W_122),
  ncalls=16, seeds=1:16)
fit12s

# GMVAR(2,2) model with mean parametrization
fit22 <- fitGMVAR(data, p=2, M=2, parametrization="mean",
  ncalls=16, seeds=1:16)
fit22

# Structural GMVAR(2,2) model with the lambda parameters restricted
# to be identical (in the second regime) and the shocks identified
# with diagonal of the B-matrix normalized positive and one zero constraint.
# The resulting model has error term covariance matrices that are
# multiplicatives of each other, while the identification equals to
# identification through Cholesky decomposition.
W_222 <- matrix(c(1, NA, 0, 1), nrow=2)
C_lambda_222 <- matrix(c(1, 1), nrow=2)
fit22s <- fitGMVAR(data, p=2, M=2, structural_pars=list(W=W_222, C_lambda=C_lambda_222),
  ncalls=20, seeds=1:20)
fit22s

# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for both regimes
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
fit22c <- fitGMVAR(data, p=2, M=2, constraints=C_mat)
fit22c

# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for both regimes and non-diagonal elements
# the coefficient matrices constrained to zero. Estimation

```

```
# with only 10 estimation rounds.
tmp <- matrix(c(1, rep(0, 10), 1, rep(0, 8), 1, rep(0, 10), 1),
  nrow=2*2^2, byrow=FALSE)
C_mat2 <- rbind(tmp, tmp)
fit22c2 <- fitGMVAR(data, p=2, M=2, constraints=C_mat2)
fit22c2
```

---

format\_valuef                      *Function factory for value formatting*

---

### Description

format\_valuef is a function factory for formatting values with certain number of digits.

### Usage

```
format_valuef(digits)
```

### Arguments

digits                      the number of decimals to print

### Value

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

---

form\_boldA                      *Form the  $((dp)x(dp))$  "bold A" matrices related to the VAR processes*

---

### Description

form\_boldA creates the "bold A" coefficient matrices related to VAR processes.

### Usage

```
form_boldA(p, M, d, all_A)
```

### Arguments

p                      a positive integer specifying the autoregressive order of the model.  
M                      a positive integer specifying the number of mixture components.  
d                      the number of time series in the system.  
all\_A                      4D array containing all coefficient matrices  $A_{m,i}$ , obtained from pick\_allA.



**Value**

Returns 3D array containing the  $((dp) \times (dp))$  "bold A" matrices related to each component VAR-process. The matrix  $A_m$  can be obtained by choosing  $[, , m]$ .

**Warning**

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

GAfit

*Genetic algorithm for preliminary estimation of a GMVAR model*


---

**Description**

GAfit estimates the specified GMVAR model using a genetic algorithm. It's designed to find starting values for gradient based methods.

**Usage**

```
GAfit(
  data,
  p,
  M,
  conditional = TRUE,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL,
  ngen = 200,
  popsize,
  smart_mu = min(100, ceiling(0.5 * ngen)),
  initpop = NULL,
  mu_scale,
  mu_scale2,
  omega_scale,
  W_scale,
  lambda_scale,
  ar_scale = 1,
  regime_force_scale = 1,
  red_criteria = c(0.05, 0.01),
```

```

to_return = c("alt_ind", "best_ind"),
minval,
seed = NULL
)

```

### Arguments

data	a matrix or class 'ts' object with $d > 1$ columns. Each column is taken to represent a single time series. NA values are not supported.
p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
conditional	a logical argument specifying whether the conditional or exact log-likelihood function
parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}$ , $m=1,\dots,M$ .
constraints	a size $(Mpd^2 \times q)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))(pd^2 \times 1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi (qx1)$ contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ $(Mpd^2 \times pd^2)$ where $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	<p>If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:</p> <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(d \times d)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• <math>C\_lambda</math> - a <math>(d(M-1) \times r)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new <math>(r \times 1)</math> parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of <math>C\_lambda</math> must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul> <p>See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is <math>W</math> times a time-varying diagonal matrix with positive diagonal entries).</p>
ngen	a positive integer specifying the number of generations to be ran through in the genetic algorithm.
popszie	a positive even integer specifying the population size in the genetic algorithm. Default is $10 \times n\_params$ .
smart_mu	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 (which has the least regimes with time varying mixing weights practically at zero) so far.

initpop

a list of parameter vectors from which the initial population of the genetic algorithm will be generated from. The parameter vectors should be...

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $\psi$  ( $qx1$ ) satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ . Here  $C$  is  $(Mpd^2 \times q)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi$  ( $qx1$ ) that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

mu\_scale

a size  $(dx1)$  vector defining **means** of the normal distributions from which each mean parameter  $\mu_m$  is drawn from in random mutations. Default is `colMeans(data)`. Note that mean-parametrization is always used for optimization in GAfit - even when parametrization=="intercept". However, input (in initpop) and output (return value) parameter vectors can be intercept-parametrized.

mu\_scale2

a size  $(dx1)$  strictly positive vector defining **standard deviations** of the normal distributions from which each mean parameter  $\mu_m$  is drawn from in random mutations. Default is `2*sd(data[, i]), i=1, . . . , d`.

omega\_scale

a size  $(dx1)$  strictly positive vector specifying the scale and variability of the random covariance matrices in random mutations. The covariance matrices are drawn from (scaled) Wishart distribution. Expected values of the random covariance matrices are `diag(omega_scale)`. Standard deviations of the diagonal elements are `sqrt(2/d)*omega_scale[i]` and for non-diagonal elements they are

- $\sqrt{1/d*\omega\_scale[i]*\omega\_scale[j]}$ ). Note that for  $d>4$  this scale may need to be chosen carefully. Default in GAfit is `var(stats::ar(data[,i],order.max=10)$resid`. This argument is ignored if structural model is considered.
- W\_scale** a size  $(dx1)$  strictly positive vector partly specifying the scale and variability of the random covariance matrices in random mutations. The elements of the matrix  $W$  are drawn independently from such normal distributions that the expectation of the main **diagonal** elements of the first regime's error term covariance matrix  $\Omega_1 = WW'$  is `W_scale`. The distribution of  $\Omega_1$  will be in some sense like a Wishart distribution but with the columns (elements) of  $W$  obeying the given constraints. The constraints are accounted for by setting the element to be always zero if it is subject to a zero constraint and for sign constraints the absolute value or negative the absolute value are taken, and then the variances of the elements of  $W$  are adjusted accordingly. This argument is ignored if reduced form model is considered.
- lambda\_scale** a length  $M - 1$  vector specifying the **standard deviation** of the mean zero normal distribution from which the eigenvalue  $\lambda_{mi}$  parameters are drawn from in random mutations. As the eigenvalues should always be positive, the absolute value is taken. The elements of `lambda_scale` should be strictly positive real numbers with the  $m - 1$ th element giving the degrees of freedom for the  $m$ th regime. The expected value of the main **diagonal** elements  $ij$  of the  $m$ th ( $m > 1$ ) error term covariance matrix will be `W_scale[i]*(d - n_i)^(-1)*sum(lambdas*ind_fun)` where the  $(dx1)$  vector `lambdas` is drawn from the absolute value of the t-distribution, `n_i` is the number of zero constraints in the  $i$ th row of  $W$  and `ind_fun` is an indicator function that takes the value one iff the  $ij$ th element of  $W$  is not constrained to zero. Basically, larger `lambdas` (or smaller degrees of freedom) imply larger variance.
- If the `lambda` parameters are **constrained** with the  $(d(M - 1)xr)$  constraint matrix `C_lambda`, then provide a length  $r$  vector specifying the standard deviation of the (absolute value of the) mean zero normal distribution each of the  $\gamma$  parameters are drawn from (the  $\gamma$  is a  $(rx1)$  vector). The expected value of the main diagonal elements of the covariance matrices then depend on the constraints.
- This argument is ignored if  $M == 1$  or a reduced form model is considered. Default is `rep(3, times=M-1)` if `lambdas` are not constrained and `rep(3, times=r)` if `lambdas` are constrained.
- As with `omega_scale` and `W_scale`, this argument should be adjusted carefully if specified by hand. **NOTE** that if `lambdas` are constrained in some other way than restricting some of them to be identical, this parameter should be adjusted accordingly in order to the estimation succeed!
- ar\_scale** a positive real number adjusting how large AR parameter values are typically generated in some random mutations. See the function `random_coefmats2` for details. This is ignored when estimating constrained models.
- regime\_force\_scale** a non-negative real number specifying how much should natural selection favour individuals with less regimes that have almost all mixing weights (practically) at zero. 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 probability weights for individuals with no wasted regimes compared to individuals with one wasted regime. Number 2 would give  $3x$  larger probability weights etc.

red_criteria	a length 2 numeric vector specifying the criteria that is used to determine whether a regime is redundant (or "wasted") or not. Any regime $m$ which satisfies $\text{sum}(\text{mixingWeights}[,m] > \text{red\_criteria}[1]) < \text{red\_criteria}[2]*n\_obs$ will be considered "redundant". One should be careful when adjusting this argument (set $c(0,0)$ to fully disable the 'redundant regime' features from the algorithm).
to_return	should the genetic algorithm return the best fitting individual which has "positive enough" mixing weights for as many 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 $-(10^{\text{ceiling}(\log_{10}(n\_obs))} + d) - 1$ .
seed	a single value, interpreted as an integer, or NULL, that sets seed for the random number generator in the beginning of the function call. If calling GAfit from fitGMVAR, use the argument seeds instead of passing the argument seed.

### Details

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

By "redundant" or "wasted" regimes we mean regimes that have the time varying mixing weights practically at zero for almost all  $t$ . A model including redundant regimes would have about the same log-likelihood value without the redundant regimes and there is no purpose to have redundant regimes in a model.

### Value

Returns the estimated parameter vector which has the form described in `initpop`.

### References

- Ansley C.F., Kohn R. 1986. A note on reparameterizing a vector autoregressive moving average model to enforce stationarity. *Journal of statistical computation and simulation*, **24**:2, 99-106.
- 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. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.

- 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.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

@export

---

get_boldA_eigens	<i>Calculate absolute values of the eigenvalues of the "bold A" matrices containing the AR coefficients</i>
------------------	---

---

### Description

get\_boldA\_eigens calculates absolute values of the eigenvalues of the "bold A" matrices containing the AR coefficients for each mixture component.

### Usage

```
get_boldA_eigens(gmvar)
```

### Arguments

gmvar            an object of class 'gmvar' created with fitGMVAR or GMVAR.

### Value

Returns a matrix with  $d * p$  rows and  $M$  columns - one column for each regime. The  $m$ th column contains the absolute values (or modulus) of the eigenvalues of the "bold A" matrix containing the AR coefficients corresponding to regime  $m$ .

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

### Examples

```
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
-0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
9.838, 0.740)
mod222 <- GMVAR(d=2, p=2, M=2, params=params222, parametrization="mean")
get_boldA_eigens(mod222)
```

---

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

---

**Description**

get\_IC calculates the information criteria values 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.

**Details**

Note that for conditional models with different autoregressive order  $p$  the information criteria values are **NOT** comparable.

**Value**

Returns a data frame containing the information criteria values.

---

get_minval	<i>Returns the default smallest allowed log-likelihood for given data.</i>
------------	--

---

**Description**

get\_minval returns the default smallest allowed log-likelihood for given data.

**Usage**

```
get_minval(data)
```

**Arguments**

data	a matrix or class 'ts' object with $d > 1$ columns. Each column is taken to represent a single time series. NA values are not supported.
------	--

**Details**

This function exists to avoid duplication inside the package.

**Value**

Returns  $-(10^{(\text{ceiling}(\log_{10}(\text{nrow}(\text{data})) + \text{ncol}(\text{data})))} - 1)$

**See Also**

[fitGMVAR](#), [GAfit](#)

---

get_omega_eigens	<i>Calculate the eigenvalues of the "Omega" error term covariance matrices</i>
------------------	--

---

**Description**

get\_omega\_eigens calculates the eigenvalues of the "Omega" error term covariance matrices for each mixture component.

**Usage**

```
get_omega_eigens(gmvar)
```

**Arguments**

gmvar            an object of class 'gmvar' created with fitGMVAR or GMVAR.

**Value**

Returns a matrix with  $d$  rows and  $M$  columns - one column for each regime. The  $m$ th column contains the eigenvalues of the "Omega" error term covariance matrix of the  $m$ th regime.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**Examples**

```
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
-0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
9.838, 0.740)
mod222 <- GMVAR(d=2, p=2, M=2, params=params222, parametrization="mean")
get_omega_eigens(mod222)
```



---

get\_regime\_autocovs     *Calculate regimewise autocovariance matrices*

---

### Description

get\_regime\_autocovs calculates the first  $p$  regimewise autocovariance matrices  $\Gamma_m(j)$  for the given GMVAR model.

### Usage

```
get_regime_autocovs(gmvar)
```

### Arguments

gmvar                    an object of class 'gmvar' created with fitGMVAR or GMVAR.

### Value

Returns an  $(dx dxp + 1xM)$  array containing the first  $p$  regimewise autocovariance matrices. The subset  $[:, j, m]$  contains the  $j-1$ :th lag autocovariance matrix of the  $m$ :th regime.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

### See Also

Other moment functions: [cond\\_moments\(\)](#), [get\\_regime\\_means\(\)](#), [uncond\\_moments\(\)](#)

### Examples

```
# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(p=1, M=2, d=2, params=params122)
get_regime_autocovs(mod122)

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.031, 2.356, 1.786, 3.000, 1.250, 0.060, 0.036,
  1.335, -0.290, -0.083, -0.047, -0.356, 0.934, -0.152, 5.201, 5.883,
  3.560, 9.799, 0.368)
```

```

mod222c <- GMVAR(p=2, M=2, d=2, params=params222c, constraints=C_mat)
get_regime_autocovs(mod222c)

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
-0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(p=2, M=2, d=2, params=params222s, structural_pars=list(W=W_222))
get_regime_autocovs(mod222s)

```

---

```
get_regime_autocovs_int
```

*Calculate regimewise autocovariance matrices*

---

## Description

get\_regime\_autocovs\_int calculates the regimewise autocovariance matrices  $\Gamma_m(j)$   $j = 0, 1, \dots, p$  for the given GMVAR model.

## Usage

```

get_regime_autocovs_int(
  p,
  M,
  d,
  params,
  constraints = NULL,
  structural_pars = NULL
)

```

## Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi$  ( $q \times 1$ ) satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times q)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi$  ( $qx1$ ) that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

constraints

a size  $(Mpd^2xq)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p})(pd^2x1), m = 1, \dots, M$ , contains the coefficient matrices and  $\psi$  ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2xpd^2$ ) where  $I = \text{diag}(p*d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new ( $rx1$ ) parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Value**

Returns an  $(dx dxp + 1xM)$  array containing the first  $p$  regimewise autocovariance matrices. The subset  $[, , j, m]$  contains the  $j-1$ :th lag autocovariance matrix of the  $m$ :th regime.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

get_regime_means	<i>Calculate regime means <math>\mu_m</math></i>
------------------	--

---

**Description**

get\_regime\_means calculates regime means  $\mu_m = (I - \sum A_{m,i})^{-1}$  for the given GMVAR model.

**Usage**

```
get_regime_means(gmvar)
```

**Arguments**

gmvar            an object of class 'gmvar' created with fitGMVAR or GMVAR.

**Value**

Returns a  $(dxM)$  matrix containing regime mean  $\mu_m$  in the  $m$ :th column,  $m = 1, \dots, M$ .

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[uncond\\_moments](#), [get\\_regime\\_autocovs](#), [cond\\_moments](#)

Other moment functions: [cond\\_moments\(\)](#), [get\\_regime\\_autocovs\(\)](#), [uncond\\_moments\(\)](#)

**Examples**

```

# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(data, p=1, M=2, params=params122)
mod122
get_regime_means(mod122)

# GMVAR(2,2), d=2 model with mean-parametrization:
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
  -0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
  0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
  9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222, parametrization="mean")
mod222
get_regime_means(mod222)

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
  -0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
  0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(data, p=2, M=2, params=params222s, structural_pars=list(W=W_222))
mod222s
get_regime_means(mod222s)

```

---

get\_regime\_means\_int    *Calculate regime means  $\mu_m$*

---

**Description**

get\_regime\_means calculates regime means  $\mu_m = (I - \sum A)^{-1}$  from the given parameter vector.

**Usage**

```

get_regime_means_int(
  p,
  M,
  d,
  params,
  parametrization = c("intercept", "mean"),
  constraints = NULL,

```

```

    structural_pars = NULL
)

```

### Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1)x1)$  and have form  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1)x1)$  and have form  $\theta= (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta= (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

parametrization

"mean" or "intercept" determining whether the model is parametrized with regime means  $\mu_m$  or intercept parameters  $\phi_{m,0}$ ,  $m=1, \dots, M$ .

constraints

a size  $(Mpd^2qx)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ ,  $m = 1, \dots, M$ , contains the

coefficient matrices and  $\psi$  ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2 \times pd^2$ ) where  $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

#### structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a ( $d \times d$ ) matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a ( $d(M - 1) \times r$ ) constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new ( $r \times 1$ ) parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the  $B$ -matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

#### Value

Returns a ( $d \times M$ ) matrix containing regime mean  $\mu_m$  in the  $m$ :th column,  $m = 1, \dots, M$ .

#### Warning

No argument checks!

#### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

get\_test\_Omega

*Compute covariance matrix Omega used in quantile residual tests*

---

#### Description

get\_test\_Omega computes the covariance matrix Omega used in the quantile residuals tests described by *Kalliovirta and Saikkonen 2010*.

**Usage**

```

get_test_Omega(
  data,
  p,
  M,
  params,
  conditional,
  parametrization,
  constraints,
  structural_pars = NULL,
  g,
  dim_g
)

```

**Arguments**

**data** a matrix or class 'ts' object with  $d > 1$  columns. Each column is taken to represent a single time series. NA values are not supported.

**p** a positive integer specifying the autoregressive order of the model.

**M** a positive integer specifying the number of mixture components.

**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (q \times 1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times q)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (q \times 1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (r \times 1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M - 1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If



parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by *Kalliovirta, Meitz and Saikkonen (2016)* introducing the GMVAR model.

conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}$ , $m=1,\dots,M$ .
constraints	a size $(Mpd^2 \times q)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2 \times 1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi$ ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ ( $Mpd^2 \times pd^2$ ) where $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	<p>If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:</p> <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(d \times d)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• <math>C\_lambda</math> - a <math>(d(M-1) \times r)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new <math>(rx1)</math> parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of <math>C\_lambda</math> must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul> <p>See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is <math>W</math> times a time-varying diagonal matrix with positive diagonal entries).</p>
g	function g specifying the transformation.
dim_g	output dimension of the transformation g.

**Value**

Returns the covariance matrix Omega described by *Kalliovirta and Saikkonen 2010*.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.

- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

```
get_unconstrained_structural_pars
```

*Get structural parameters that indicate there are no constraints*

---

### Description

get\_unconstrained\_struct\_pars return structural parameters that indicate there are no constraints (except possibly sign constraints).

### Usage

```
get_unconstrained_structural_pars(structural_pars = NULL)
```

### Arguments

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the  $B$ -matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Details

Intended to be called after calling the function reform\_constrained\_pars to avoid remove the constraints again in any further function calls as this will create bugs. Sign constraints are irrelevant in this context.

### Value

Returns a list with  $W$  being  $(d \times d)$  matrix of ones and  $C\_lambda$  being NULL. If the supplied argument is NULL, returns NULL.

### Warning

No argument checks!

---

GIRF	<i>Estimate generalized impulse response function for a structural GM-VAR model.</i>
------	--

---

### Description

GIRF estimate generalized impulse response function for a structural GMVAR model.

### Usage

```
GIRF(
  gmvar,
  variables,
  shock_size,
  N = 30,
  R1 = 250,
  R2 = 250,
  init_regimes = 1:M,
  init_values = NULL,
  which_cumulative,
  ci = c(0.95, 0.8),
  include_mixweights = TRUE,
  ncores = min(2, parallel::detectCores()),
  seeds = NULL
)

## S3 method for class 'girf'
plot(x, add_grid = FALSE, ...)

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

### Arguments

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
variables	a numeric vector of length at most $d$ (=ncol(data)) and elements in $1, \dots, d$ specifying the variables for which the GIRF should be estimated.
shock_size	a vector with the same length as variables specifying the size of the structural shock for each variable. Alternatively, is a scalar value that specifies a common shock size for all variables. By default, the shock size is one, which is then amplified by the B-matrix according to the conditional standard deviation of the model.
N	a positive integer specifying the horizon how far ahead should the generalized impulse responses be calculated?
R1	the number of repetitions used to estimate GIRF for each initial value?

R2	the number of initial values to be drawn from a stationary distribution of the process or of a specific regime? The confidence bounds will be sample quantiles of the GIRFs based on different initial values. Ignored if the argument <code>init_value</code> is specified.
<code>init_regimes</code>	a numeric vector of length at most $M$ and elements in $1, \dots, M$ specifying the regimes from which the initial values should be generated from. The initial values will be generated from a mixture distribution with the mixture components being the stationary distributions of the specific regimes and the (proportional) mixing weights given by the mixing weight parameters of those regimes. Note that if <code>init_regimes=1:M</code> , the initial values are generated from the stationary distribution of the process and if <code>init_regimes=m</code> , the initial value are generated from the stationary distribution of the $m$ th regime. Ignored if <code>init_value</code> is specified.
<code>init_values</code>	a matrix or a multivariate class 'ts' object with $d$ columns and at least $p$ rows specifying an initial value for the GIRF. The last $p$ rows are taken to be the initial value assuming that the <b>last</b> row is the most recent observation.
<code>which_cumulative</code>	a numeric vector with values in $1, \dots, d$ ( $d = \text{ncol}(\text{data})$ ) specifying which the variables for which the impulse responses should be cumulative. Default is none.
<code>ci</code>	a numeric vector with elements in $(0, 1)$ specifying the confidence levels of the confidence intervals.
<code>include_mixweights</code>	should the generalized impulse response be calculated for the mixing weights as well? TRUE or FALSE.
<code>ncores</code>	the number CPU cores to be used in parallel computing. Only single core computing is supported if an initial value is specified (and the GIRF won't thus be estimated multiple times).
<code>seeds</code>	a length R2 vector containing the random number generator seed for estimation of each GIRF. A single number of an initial value is specified. or NULL for not initializing the seed. Exists for creating reproducible results.
<code>x</code>	object of class 'girf' generated by the function GIRF.
<code>add_grid</code>	should grid be added to the plots?
<code>...</code>	arguments passed to <code>grid</code> which plots grid to the figure.
<code>digits</code>	the number of decimals to print
<code>N_to_print</code>	an integer specifying the horizon how far to print the estimates and confidence intervals. The default is that all the values are printed ( <code>N_to_print=N</code> ).

### Details

The model needs to be structural in order for this function to be applicable. A structural GMVAR model can be estimated by specifying the argument `structural_pars` in the function `fitGMVAR`.

The confidence bounds reflect uncertainty about the initial state (but currently not about the parameter estimates) if initial values are not specified. If initial values are specified, there won't currently be confidence intervals. See the cited paper by Virolainen (2020) for details about the algorithm.

**Value**

Returns a class 'girf' list with the GIRFs in the first element (`$girf_res`) and the used arguments the rest. The first element containing the GIRFs is a list with the  $m$ th element containing the point estimates for the GIRF in `$point_est` (the first element) and confidence intervals in `$conf_ints` (the second element). The first row is for the GIRF at impact ( $n = 0$ ), the second for  $n = 1$ , the third for  $n = 2$ , and so on.

**Methods (by generic)**

- `plot`: plot method
- `print`: print method

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [gmvar\\_to\\_sgmvar](#), [reorder\\_W\\_columns](#), [swap\\_W\\_signs](#), [simulateGMVAR](#), [predict.gmvar](#), [profile\\_logliks](#), [quantile\\_residual\\_tests](#), [LR\\_test](#), [Wald\\_test](#)

**Examples**

```
# These are long-running examples that use parallel computing.
data(eurusd, package="gmvarkit")
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.428, -0.808, 1.029, 5.84, 1.314, 0.145, 0.094, 1.292,
  -0.389, -0.07, -0.109, -0.281, 1.248, 0.077, -0.04, 1.266, -0.272,
  -0.074, 0.034, -0.313, 0.903, 0.718, -0.324, 2.079, 7, 1.44, 0.742)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(data, p=2, M=2, params=params222s, structural_pars=list(W=W_222))
mod222s

# Alternatively, use:
# fit222s <- fitGMVAR(data, p=2, M=2, structural_pars=list(W=W_222),
#   ncalls=20, seeds=1:20)
# To obtain an estimated version of the same model.

# Estimating the GIRFs of both variables with default arguments
# (initial values are drawn from the stationary distribution of the process,
# 30 periods ahead, confidence levels 0.95 and 0.8):
girf1 <- GIRF(mod222s)
plot(girf1)
girf1
```

```

# Estimating the GIRF of the second variable only, 36 periods ahead
# and shock size 1, initial values drawn from the stationary distribution
# of the first regime, confidence level 0.9:
girf2 <- GIRF(mod222s, variables=2, shock_size=1, N=36, init_regimes=1, ci=0.9)
plot(girf2)

# Estimating the GIRFs of both variables, shock sizes 1 and 3, N=50 periods ahead,
# estimation based on 1000 Monte Carlo simulations, and fixed initial values given
# by the last p observations of the data:
girf3 <- GIRF(mod222s, shock_size=c(1, 3), N=50, R1=1000, init_values=mod222s$data)
plot(girf3)

```

---

GMVAR	<i>Create a class 'gmvar' object defining a reduced form or structural GMVAR model</i>
-------	--

---

## Description

GMVAR creates a class 'gmvar' object that defines a reduced form or structural GMVAR model

## Usage

```

GMVAR(
  data,
  p,
  M,
  d,
  params,
  conditional = TRUE,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL,
  calc_cond_moments,
  calc_std_errors = FALSE
)

```

```

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

```

```

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

```

```

## S3 method for class 'gmvar'
summary(object, ..., digits = 2)

```

```

## S3 method for class 'gmvar'

```

```

plot(x, ...)

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

```

### Arguments

**data** a matrix or class 'ts' object with  $d > 1$  columns. Each column is taken to represent a single times series. NA values are not supported. Ignore if defining a model without data is desired.

**p** a positive integer specifying the autoregressive order of the model.

**M** a positive integer specifying the number of mixture components.

**d** number of times series in the system, i.e. `ncol(data)`. This can be used to define GMVAR models without data and can be ignored if data is provided.

**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times q)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M - 1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If `parametrization=="mean"`, just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ . `vec()` is vectorization operator that stacks columns of a given matrix into a vector. `vech()` stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by *Kalliovirta, Meitz and Saikkonen (2016)* introducing the GMVAR model.

conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}$ , $m=1,\dots,M$ .
constraints	a size $(Mpd^2 \times q)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))(pd^2 \times 1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi$ ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ ( $Mpd^2 \times pd^2$ ) where $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	If NULL a reduced form model is considered. For structural model, should be a list containing the following elements: <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(d \times d)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• <math>C_\lambda</math> - a <math>(d(M-1) \times r)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new <math>(rx1)</math> parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of <math>C_\lambda</math> must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul> <p>See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is <math>W</math> times a time-varying diagonal matrix with positive diagonal entries).</p>
calc_cond_moments	should conditional means and covariance matrices should be calculated? Default is TRUE if the model contains data and FALSE otherwise.
calc_std_errors	should approximate standard errors be calculated?
object	object of class 'gmvar' generated by fitGMVAR or GMVAR.
...	correctly not used.
digits	number of digits to be printed.
x	object of class 'gmvar' generated by fitGMVAR or GMVAR.
summary_print	if set to TRUE then the print will include log-likelihood and information criteria values.

## Details

If data is provided, then also multivariate quantile residuals (*Kalliovirta and Saikkonen 2010*) are computed and included in the returned object.

The first plot displays the time series together with estimated mixing weights. The second plot displays (Gaussian) kernel density estimates of the individual series together with the marginal stationary density implied by the model. The colored regimewise stationary densities are multiplied with the mixing weight parameter estimates.



**Value**

Returns an object of class 'gmvar' defining the specified reduced form or structural GMVAR model. Can be used to work with other functions provided in gmvarKit.

Remark that the first autocovariance/correlation matrix in \$uncond\_moments is for the lag zero, the second one for the lag one, etc.

**Methods (by generic)**

- logLik: Log-likelihood method
- residuals: residuals method to extract multivariate quantile residuals
- summary: summary method
- plot: plot method for class 'gmvar'
- print: print method

**About S3 methods**

Only the print method is available if data is not provided. If data is provided, then in addition to the ones listed above, the predict method is also available.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [add\\_data](#), [swap\\_parametrization](#), [GIRF](#), [gmvar\\_to\\_sgmvar](#), [reorder\\_W\\_columns](#), [swap\\_W\\_signs](#)

**Examples**

```
# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
3.982, 12.135, 0.789)
mod122 <- GMVAR(data, p=1, M=2, params=params122)
mod122

# GMVAR(1,2), d=2 model without data
```

```

mod122_2 <- GMVAR(p=1, M=2, d=2, params=params122)
mod122_2

# GMVAR(2,2), d=2 model with mean-parametrization:
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
-0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222, parametrization="mean")
mod222

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(-11.964, 155.024, 11.636, 124.988, 1.314, 0.145, 0.094, 1.292,
-0.389, -0.07, -0.109, -0.281, 1.248, 0.077, -0.04, 1.266, -0.272, -0.074,
0.034, -0.313, 0.903, 0.718, -0.324, 2.079, 7.00, 1.44, 0.742)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(data, p=2, M=2, params=params222s, parametrization="mean",
structural_pars=list(W=W_222))
mod222s

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.031, 2.356, 1.786, 3.000, 1.250, 0.060, 0.036,
1.335, -0.290, -0.083, -0.047, -0.356, 0.934, -0.152, 5.201, 5.883,
3.560, 9.799, 0.368)
mod222c <- GMVAR(data, p=2, M=2, params=params222c, constraints=C_mat)
mod222c

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes and the non-diagonal elements of
# the coefficient matrices constrained to zero.
tmp <- matrix(c(1, rep(0, 10), 1, rep(0, 8), 1, rep(0, 10), 1),
nrow=2*2^2, byrow=FALSE)
C_mat2 <- rbind(tmp, tmp)
params222c2 <- c(0.355, 3.193, -0.114, 2.829, 1.263, 1.338, -0.292,
-0.362, 5.597, 3.456, 9.622, 0.982, -0.327, 5.236, 0.650)
mod222c2 <- GMVAR(data, p=2, M=2, params=params222c2,
constraints=C_mat2)
mod222c2

```

---

gmvarkit

*gmvarkit: Estimate Gaussian Mixture Vector Autoregressive (GM-  
VAR) model*


---

## Description

gmvarkit is a package for reduced form and structural Gaussian mixture vector autoregressive (GMVAR) model analysis. It provides functions for unconstrained and constrained maximum likelihood estimation of the model parameters, quantile residuals tests, graphical diagnostics, estima-

tion of generalized impulse response function, simulation from GMVAR processes, forecasting, and more.

Many of the functions documented are not exported but intended internal use only. The readme file is a good place to start and the vignette might be useful too.

---

gmvar_to_sgmvar	<i>Switch from two-regime reduced form GMVAR model to a structural GMVAR model.</i>
-----------------	---

---

### Description

gmvar\_to\_sgmvar constructs SGMVAR model based on a reduced form GMVAR model.

### Usage

```
gmvar_to_sgmvar(gmvar)
```

### Arguments

gmvar            an object of class 'gmvar' created with fitGMVAR or GMVAR.

### Details

The switch is made by simultaneously diagonalizing the two error term covariance matrices with a well known matrix decomposition (Muirhead, 1982, Theorem A9.9) and then normalizing the diagonal of the matrix  $W$  positive (which implies positive diagonal of the B-matrix). Models with more than two regimes are not supported because the matrix decomposition does not generally exist for more than two covariance matrices. If the model has only one regime (= regular SVAR model), a symmetric and pos. def. square root matrix of the error term covariance matrix is used.

The columns of  $W$  as well as the lambda parameters can be re-ordered (without changing the implied reduced form model) afterwards with the function reorder\_W\_columns. Also all signs in any column of  $W$  can be swapped (without changing the implied reduced form model) afterwards with the function swap\_W\_signs. These two functions work with models containing any number of regimes.

### Value

Returns an object of class 'gmvar' defining a structural GMVAR model based on a two-regime reduced form GMVAR model with the main diagonal of the B-matrix normalized to be positive.

### References

- Muirhead R.J. 1982. Aspects of Multivariate Statistical Theory, Wiley.
- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [GIRF](#), [reorder\\_W\\_columns](#), [swap\\_W\\_signs](#)

**Examples**

```
# These are long running examples
data(eurusd, package="gmvarkit")
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# Reduced form GMVAR(1,2) model
fit12 <- fitGMVAR(data, p=1, M=2, ncalls=1, seeds=3)

# Form a structural model based on the reduced form model:
mod12s <- gmvar_to_sgmvar(fit12)
```

---

in\_paramspace

*Determine whether the parameter vector lies in the parameter space*


---

**Description**

in\_paramspace checks whether the given parameter vector lies in the parameter space. Does NOT consider the identifiability condition!

**Usage**

```
in_paramspace(p, M, d, params, constraints = NULL, structural_pars = NULL)
```

**Arguments**

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi$  ( $qx1$ ) that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

constraints

a size  $(Mpd^2xq)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))(pd^2x1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi$  ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2xpd^2$ ) where  $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new ( $rx1$ ) parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

## Value

Returns TRUE if the given parameter vector lies in the parameter space and FALSE otherwise.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

## Examples

```
# GMVAR(1,1), d=2 model:
params112 <- c(1.07, 127.71, 0.99, 0.00, -0.01, 0.99, 4.05,
  2.22, 8.87)
in_paramspace(p=1, M=1, d=2, params=params112)

# GMVAR(2,2), d=2 model:
params222 <- c(1.39, -0.77, 1.31, 0.14, 0.09, 1.29, -0.39,
  -0.07, -0.11, -0.28, 0.92, -0.03, 4.84, 1.01, 5.93, 1.25,
  0.08, -0.04, 1.27, -0.27, -0.07, 0.03, -0.31, 5.85, 3.57,
  9.84, 0.74)
in_paramspace(p=2, M=2, d=2, params=params222)

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.03, 2.36, 1.79, 3.00, 1.25, 0.06, 0.04,
  1.34, -0.29, -0.08, -0.05, -0.36, 0.93, -0.15, 5.20,
  5.88, 3.56, 9.80, 0.37)
in_paramspace(p=2, M=2, d=2, params=params222c, constraints=C_mat)

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
  -0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
  0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
in_paramspace(p=2, M=2, d=2, params=params222s,
  structural_pars=list(W=W_222))
```

---

<code>in_paramspace_int</code>	<i>Determine whether the parameter vector lies in the parameter space</i>
--------------------------------	---

---

## Description

`in_paramspace_int` checks whether the parameter vector lies in the parameter space.

## Usage

```
in_paramspace_int(
  p,
  M,
```

```

    d,
    params,
    all_boldA,
    alphas,
    all_Omega,
    W_constraints = NULL
)

```

### Arguments

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.
params	a real valued vector specifying the parameter values.
	<p><b>For reduced form model:</b> Should be size <math>((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)</math> and have form <math>\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})</math>, where:</p> <ul style="list-style-type: none"> <li>• <math>v_m = (\phi_{m,0}, \phi_m, \sigma_m)</math></li> <li>• <math>\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))</math></li> <li>• and <math>\sigma_m = vech(\Omega_m)</math>, <math>m=1, \dots, M</math>.</li> </ul> <p><b>For structural GMVAR model:</b> Should have the form <math>\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)</math> where</p> <ul style="list-style-type: none"> <li>• <math>\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})</math> contains the eigenvalues of the <math>m</math>th mixture component.</li> </ul> <p>Above, <math>\phi_{m,0}</math> is the intercept parameter, <math>A_{m,i}</math> denotes the <math>i</math>:th coefficient matrix of the <math>m</math>:th mixture component, <math>\Omega_m</math> denotes the error term covariance matrix of the <math>m</math>:th mixture component, and <math>\alpha_m</math> is the mixing weight parameter. The <math>W</math> and <math>\lambda_{mi}</math> are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If <math>M = 1</math>, <math>\alpha_m</math> and <math>\lambda_{mi}</math> are dropped.</p> <p>If parametrization=="mean", just replace each <math>\phi_{m,0}</math> with the regimewise mean <math>\mu_m</math>. <math>vec()</math> is vectorization operator that stacks columns of a given matrix into a vector. <math>vech()</math> stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.</p>
all_boldA	3D array containing the $((dp) \times (dp))$ "bold A" matrices related to each mixture component VAR-process, obtained from form_boldA. Will be computed if not given.
alphas	(Mx1) vector containing all mixing weight parameters, obtained from pick_alphas.
all_Omega	3D array containing all covariance matrices $\Omega_m$ , obtained from pick_Omegas.
W_constraints	set NULL for reduced form models. For structural models, this should be the constraint matrix $W$ from the list of structural parameters.

### Details

The parameter vector in the argument params should be unconstrained and it is used for structural models only.

**Value**

Returns TRUE if the given parameter values are in the parameter space and FALSE otherwise. This function does NOT consider the identifiability condition!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

is\_stationary

---

*Check the stationary condition of a given GMVAR model*


---

**Description**

is\_stationary checks the stationarity condition of a GMVAR model.

**Usage**

```
is_stationary(
  p,
  M,
  d,
  params,
  all_boldA = NULL,
  structural_pars = NULL,
  tolerance = 0.001
)
```

**Arguments**

p a positive integer specifying the autoregressive order of the model.  
M a positive integer specifying the number of mixture components.  
d the number of time series in the system.  
params a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.



Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If `parametrization=="mean"`, just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ . `vec()` is vectorization operator that stacks columns of a given matrix into a vector. `vech()` stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

`all_boldA` 3D array containing the  $((dp)x(dp))$  "bold A" matrices related to each mixture component VAR-process, obtained from `form_boldA`. Will be computed if not given.

`structural_pars`

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- `W` - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- `C_lambda` - a  $(d(M-1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of `C_lambda` must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

`tolerance` Returns FALSE if modulus of any eigenvalue is larger or equal to  $1 - \text{tolerance}$ .

## Details

If the model is constrained, remove the constraints first with the function `reform_constrained_pars`.

## Value

Returns TRUE if the model is stationary and FALSE if not. Based on the argument `tolerance`, `is_stationary` may return FALSE when the parameter vector is in the stationarity region, but very close to the boundary (this is used to ensure numerical stability in estimation of the model parameters).

## Warning

No argument checks!

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

iterate_more	<i>Maximum likelihood estimation of a GMVAR model with preliminary estimates</i>
--------------	--

---

## Description

iterate\_more uses a variable metric algorithm to finalize maximum likelihood estimation of a GMVAR model (object of class 'gmvar') which already has preliminary estimates.

## Usage

```
iterate_more(gmvar, maxit = 100, calc_std_errors = TRUE)
```

## Arguments

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
maxit	the maximum number of iterations in the variable metric algorithm.
calc_std_errors	should approximate standard errors be calculated?

## Details

The 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 GMVAR model with the main estimation function fitGMVAR. iterate\_more is essentially a wrapper around the function optim from the package stats and GMVAR from the package gmvarKit.

## Value

Returns an object of class 'gmvar' defining the estimated GMVAR model.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [optim](#), [profile\\_logliks](#)

**Examples**

```
## These are long running examples that use parallel computing!

# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2) model, only 5 iterations of the variable metric
# algorithm
fit12 <- fitGMVAR(data, p=1, M=2, maxit=5)
fit12

# Iterate more:
fit12_2 <- iterate_more(fit12)
fit12_2

# Structural GMVAR(1,2) model identified with sign
# constraints. Only 10 iterations of the variable metric
# algorithm
W_122 <- matrix(c(1, -1, 1, 1), nrow=2)
fit12s <- fitGMVAR(data, p=1, M=2, structural_pars=list(W=W_122),
  ncalls=16, maxit=10, seeds=1:16)
fit12s

# Iterate more:
fit12s_2 <- iterate_more(fit12s)
fit12s_2

# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for all regimes, only 10 iterations of the
# variable metric algorithm
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
fit22c <- fitGMVAR(data, p=2, M=2, constraints=C_mat, maxit=10)
fit22c

# Iterate more:
fit22c_2 <- iterate_more(fit22c)
fit22c_2

# GMVAR(3,2) model, only 10 iterations of the variable metric
# algorithm
fit32 <- fitGMVAR(data, p=3, M=2, maxit=10)
fit32

# Iterate more:
fit32_2 <- iterate_more(fit32)
```

fit32\_2

loglikelihood

*Compute log-likelihood of a GMVAR model using parameter vector***Description**

loglikelihood computes log-likelihood of a GMVAR model using parameter vector instead of an object of class 'gmvar'. Exists for convenience if one wants to for example employ other estimation algorithms than the ones used in fitGMVAR. Use minval to control what happens when the parameter vector is outside the parameter space.

**Usage**

```
loglikelihood(
  data,
  p,
  M,
  params,
  conditional = TRUE,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL,
  minval = NA
)
```

**Arguments**

**data** a matrix or class 'ts' object with  $d > 1$  columns. Each column is taken to represent a single time series. NA values are not supported.

**p** a positive integer specifying the autoregressive order of the model.

**M** a positive integer specifying the number of mixture components.

**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1)x1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1)x1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2qxq)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi$  ( $qx1$ ) that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}$ , $m=1, \dots, M$ .
constraints	a size $(Mpd^2xq)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi$ ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ ( $Mpd^2xpd^2$ ) where $I = \text{diag}(p*d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	If NULL a reduced form model is considered. For structural model, should be a list containing the following elements: <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(dxd)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• <math>C\_lambda</math> - a <math>(d(M-1)xr)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new (<math>rx1</math>) parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of <math>C\_lambda</math> must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul>

See Virolainen (2020) for the conditions required to identify the shocks and for the  $B$ -matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

`minval` the value that will be returned if the parameter vector does not lie in the parameter space (excluding the identification condition).

### Details

`loglikelihood_int` takes use of the function `dmvn` from the package `mvnfast` to cut down computation time. Values extremely close to zero are handled with the package `Brobdingnag`.

### Value

Returns log-likelihood if `params` is in the parameters space and `minval` if not.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

### See Also

[fitGMVAR](#), [GMVAR](#), [calc\\_gradient](#)

### Examples

```
data <- cbind(10*eurusd[,1], 100*eurusd[,2])

# GMVAR(2, 2), d=2 model;
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
-0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
9.838, 0.740)
loglikelihood(data=data, p=2, M=2, params=params222, parametrization="mean")

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
-0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
loglikelihood(data=data, p=2, M=2, params=params222s, structural_pars=list(W=W_222))
```

---

loglikelihood_int	<i>Compute log-likelihood of a Gaussian mixture vector autoregressive model</i>
-------------------	---

---

### Description

loglikelihood\_int computes log-likelihood of a GMVAR model.

### Usage

```
loglikelihood_int(
  data,
  p,
  M,
  params,
  conditional = TRUE,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL,
  to_return = c("loglik", "mw", "mw_tplus1", "loglik_and_mw", "terms", "regime_cmeans",
    "total_cmeans", "total_ccovs"),
  check_params = TRUE,
  minval = NULL
)
```

### Arguments

data	a matrix or class 'ts' object with $d > 1$ columns. Each column is taken to represent a single time series. NA values are not supported.
p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
params	a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi$  ( $qx1$ ) that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regime-wise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}$ , $m=1, \dots, M$ .
constraints	a size $(Mpd^2xq)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi$ ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ ( $Mpd^2xpd^2$ ) where $I = \text{diag}(p \cdot d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	If NULL a reduced form model is considered. For structural model, should be a list containing the following elements: <ul style="list-style-type: none"> <li>• W - a <math>(dxd)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• C_lambda - a <math>(d(M-1)xr)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new (<math>rx1</math>) parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of C_lambda must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul> <p>See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is <math>W</math> times a time-varying diagonal matrix with positive diagonal entries).</p>
to_return	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 *KMS 2016, eq.(9)*)? Or should the regimewise conditional means, total conditional means, or total conditional covariance matrices be returned? Default is the log-likelihood value ("loglik").

check_params	should it be checked that the parameter vector satisfies the model assumptions? Can be skipped to save computation time if it does for sure.
minval	the value that will be returned if the parameter vector does not lie in the parameter space (excluding the identification condition).

### Details

loglikelihood\_int takes use of the function dmvn from the package mvnfast to cut down computation time. Values extremely close to zero are handled with the package Brobdingnag.

### Value

**By default:** log-likelihood value of the specified GMVAR 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.

**If to\_return=="regime\_cmeans":** an  $[T-p, d, M]$  array containing the regimewise conditional means (the first p values are used as the initial values).

**If to\_return=="total\_cmeans":** a  $[T-p, d]$  matrix containing the conditional means of the process (the first p values are used as the initial values).

**If to\_return=="total\_ccov":** an  $[d, d, T-p]$  array containing the conditional covariance matrices of the process (the first p values are used as the initial values).

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

LR\_test

*Perform likelihood ratio test for a GMVAR or SGMVAR model***Description**

LR\_test performs a likelihood ratio test for a GMVAR or SGMVAR model

**Usage**

```
LR_test(gmvar1, gmvar2)

## S3 method for class 'lr'
print(x, ..., digits = 4)
```

**Arguments**

gmvar1	an object of class 'gmvar' generated by fitGMVAR or GMVAR, containing the <b>freely estimated</b> model.
gmvar2	an object of class 'gmvar' generated by fitGMVAR or GMVAR, containing the <b>constrained</b> model.
x	object of class 'lr' generated by the function LR_test.
...	currently not used.
digits	how many significant digits to print?

**Details**

Performs a likelihood ratio test, testing the null hypothesis that the true parameter value lies in the constrained parameter space. Under the null, the test statistic is asymptotically  $\chi^2$ -distributed with  $k$  degrees of freedom,  $k$  being the difference in the dimensions of the unconstrained and constrained parameter spaces.

Note that this function does **not** verify that the two models are actually nested.

**Value**

Returns an object of class 'lr' containing the test statistic and the related p-value.

**Methods (by generic)**

- print: print method

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[Wald\\_test](#), [fitGMVAR](#), [GMVAR](#), [diagnostic\\_plot](#), [profile\\_logliks](#), [quantile\\_residual\\_tests](#), [cond\\_moment\\_plot](#)

**Examples**

```
# Load the data
data(eurusd, package="gmvarkit")
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# Structural GMVAR(2, 2), d=2 model identified similarly to Cholesky:
W_222 <- matrix(c(1, NA, 0, 1), nrow=2, byrow=FALSE)
fit222s <- fitGMVAR(data, p=2, M=2, structural_pars=list(W=W_222),
                   ncalls=1, seeds=16)

# The same model but the AR coefficients restricted to be the same
# in both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
fit222sc <- fitGMVAR(data, p=2, M=2, constraints=C_mat,
                    structural_pars=list(W=W_222),
                    ncalls=1, seeds=16)

# Test whether the constraints are supported by the data:
LR_test(fit222s, fit222sc)
```

---

n_params	<i>Calculate the number of parameters in GMVAR model parameter vector</i>
----------	---

---

**Description**

n\_params calculates the number of parameters in the model.

**Usage**

```
n_params(p, M, d, constraints = NULL, structural_pars = NULL)
```

**Arguments**

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.

constraints a size  $(Mpd^2xq)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi$  ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2xpd^2$ ) where  $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Value

Returns the number of parameters in parameter vector of the specified GMVAR model.

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick\_allA

*Pick coefficient all matrices*

---

### Description

pick\_allA picks all coefficient matrices  $A_{m,i}$  ( $i = 1, \dots, p$ ,  $m = 1, \dots, M$ ) from the given parameter vector so that they are arranged in a 4D array with the fourth dimension indicating each component and third dimension indicating each lag.

**Usage**

```
pick_allA(p, M, d, params, structural_pars = NULL)
```

**Arguments**

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If parametrization=="mean", just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

**structural\_pars**

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- **W** - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- **C\_lambda** - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of **C\_lambda** must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Value**

Returns a 4D array containing the coefficient matrices of the all components. Coefficient matrix  $A_{m,i}$  can be obtained by choosing  $[, , i, m]$ .

**Warning**

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick_all_phi0_A	<i>Pick all <math>\phi_{m,0}</math> or <math>\mu_m</math> and <math>A_{m,1}, \dots, A_{m,p}</math> parameter values</i>
-----------------	---

---

**Description**

pick\_all\_phi0\_A picks the intercept or mean parameters and vectorized coefficient matrices from the given parameter vector.

**Usage**

```
pick_all_phi0_A(p, M, d, params, structural_pars = NULL)
```

**Arguments**

- p a positive integer specifying the autoregressive order of the model.  
M a positive integer specifying the number of mixture components.  
d the number of time series in the system.  
params a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If `parametrization=="mean"`, just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ . `vec()` is vectorization operator that stacks columns of a given matrix into a vector. `vech()` stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

#### structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- `W` - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- `C_lambda` - a  $(d(M-1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of `C_lambda` must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

#### Value

Returns a  $((pd^2 + d) \times M)$  matrix containing  $(\phi_{m,0}, \text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$  in the  $m$ :th column, or  $(\mu_m, \text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$  if the parameter vector is mean-parametrized,  $m=1, \dots, M$ .

#### Warning

No argument checks!

#### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick_alphas	<i>Pick mixing weight parameters <math>\alpha_m, m = 1, \dots, M</math></i>
-------------	---

---

### Description

pick\_alphas picks the mixing weight parameters from the given parameter vector.

### Usage

```
pick_alphas(p, M, d, params)
```

### Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m), m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If `parametrization=="mean"`, just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ . `vec()` is vectorization operator that stacks columns of a given matrix into a vector. `vech()` stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

### Value

Returns a length  $M$  vector containing the mixing weight parameters  $alpha_m, m = 1, \dots, M$ , including non-parametrized  $alpha_M$ .

### Warning

No argument checks!



## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick\_Am *Pick coefficient matrices*

---

## Description

pick\_Am picks the coefficient matrices  $A_{m,i}$  ( $i = 1, \dots, p$ ) from the given parameter vector so that they are arranged in a 3D array with the third dimension indicating each lag.

## Usage

```
pick_Am(p, M, d, params, m, structural_pars = NULL)
```

## Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** number of time series in the system, i.e. the dimension.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi$  ( $qx1$ ) satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi$  ( $qx1$ ) that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

m which component?

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Value

Returns a 3D array containing the coefficient matrices of the given component. A coefficient matrix  $A_{m,i}$  can be obtained by choosing  $[, , i]$ .

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick_Ami	<i>Pick coefficient matrix</i>
----------	--------------------------------

---

### Description

pick\_Ami picks the coefficient matrix  $A_{m,i}$  from the given parameter vector.

### Usage

```
pick_Ami(p, M, d, params, m, i, structural_pars = NULL, unvec = TRUE)
```

### Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** number of time series in the system, i.e. the dimension.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .

`vec()` is vectorization operator that stacks columns of a given matrix into a vector. `vech()` stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by *Kalliovirta, Meitz and Saikkonen (2016)* introducing the GMVAR model.

m	which component?
i	which lag in 1,...,p?
structural_pars	<p>If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:</p> <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(d \times d)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• <math>C\_lambda</math> - a <math>(d(M - 1) \times r)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new <math>(r \times 1)</math> parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of <math>C\_lambda</math> must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul> <p>See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is <math>W</math> times a time-varying diagonal matrix with positive diagonal entries).</p>
unvec	if FALSE then vectorized version of $A_{m,i}$ will be returned instead of matrix. Default if TRUE.

### Details

Does not support constrained parameter vectors.

### Value

Returns the  $i$ :th lag coefficient matrix of  $m$ :th component,  $A_{m,i}$ .

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. *New Introduction to Multiple Time Series Analysis*, Springer.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

pick\_lambdas

*Pick the structural parameters eigenvalue 'lambdas'***Description**

pick\_lambdas picks the structural parameters eigenvalue 'lambdas' from a parameter vector

**Usage**

```
pick_lambdas(p, M, d, params, structural_pars = NULL)
```

**Arguments**

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If parametrization=="mean", just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.

- C\_lambda - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of C\_lambda must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Details

Constrained parameter vectors are not supported. Not even constraints in  $W$ !

### Value

Returns a length  $(d * (M - 1))$  vector  $(\lambda_2, \dots, \lambda_M)$  (see the argument params) from a parameter vector of a SGMVAR model. Returns `numeric(0)` for reduced form models or when  $M = 1$ .

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick\_Omegas

*Pick covariance matrices*

---

### Description

pick\_Omegas picks the covariance matrices  $\Omega_m (m = 1, \dots, M)$  from the given parameter vector so that they are arranged in a 3D array with the third dimension indicating each component.

### Usage

```
pick_Omegas(p, M, d, params, structural_pars = NULL)
```

**Arguments**

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.
params	a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If parametrization=="mean", just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Value**

Returns a 3D array containing the covariance matrices of the given model. Coefficient matrix  $\Omega_m$  can be obtained by choosing  $[, , m]$ .

**Warning**

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick_phi0	<i>Pick <math>\phi_{m,0}</math> or <math>\mu_{m,0}</math>, <math>m=1,\dots,M</math> vectors</i>
-----------	---

---

**Description**

pick\_phi0 picks the intercept or mean parameters from the given parameter vector.

**Usage**

```
pick_phi0(p, M, d, params, structural_pars = NULL)
```

**Arguments**

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1,\dots,M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If `parametrization=="mean"`, just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ . `vec()` is vectorization operator that stacks columns of a given matrix into a vector. `vech()` stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.



## structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the  $B$ -matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Value**

Returns a  $(d \times M)$  matrix containing  $\phi_{m,0}$  in the  $m$ :th column or  $\mu_m$  if the parameter vector is mean-parametrized,  $m = 1, \dots, M$ .

**Warning**

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick\_regime

*Pick regime parameters  $v_m = (\phi_m, 0, \phi_m, \sigma_m)$*

---

**Description**

pick\_regime picks the regime-parameters from the given parameter vector.

**Usage**

pick\_regime(p, M, d, params, m, structural\_pars = NULL)

**Arguments**

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	number of time series in the system, i.e. the dimension.
params	a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1)x1)$  and have form  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1)x1)$  and have form  $\theta= (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta= (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If `parametrization="mean"`, just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by *Kalliovirta, Meitz and Saikkonen (2016)* introducing the GMVAR model.

m which component?

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.

- C\_lambda - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of C\_lambda must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Details

Models with AR or lambda parameter constraints are currently not supported.

### Value

**For reduced form models:** returns length  $pd^2 + d + d(d+1)/2$  vector containing  $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$ , where  $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,1}))$  and  $\sigma_m = vech(\Omega_m)$ .

**For structural models:** returns the length  $pd^2 + d$  vector  $(\phi_{m,0}, \phi_m)$ .

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

pick\_W

*Pick the structural parameter matrix W*

---

### Description

pick\_W picks the structural parameter matrix W from a parameter vector

### Usage

pick\_W(p, M, d, params, structural\_pars = NULL)

## Arguments

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.
params	a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If parametrization=="mean", just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- W - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- C\_lambda - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of C\_lambda must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

## Details

Constrained parameter vectors are not supported. Not even constraints in  $W$ !

**Value**

Returns a  $(d \times d)$  matrix  $W$  from a parameter vector of a SGMVAR model. Returns NULL for reduced form models.

**Warning**

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

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

---

**Description**

plot.gmvarpred is plot method for gmvarpred objects.

**Usage**

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

**Arguments**

x	object of class 'gmvarpred' generated by predict.gmvar.
...	arguments passed to grid which plots grid to the figure.
nt	a positive integer specifying the number of observations to be plotted along with the prediction (ignored if plot_res==FALSE). Default is round(nrow(data)*0.15).
mix_weights	TRUE if forecasts for mixing weights should be plotted, FALSE in not.
add_grid	should grid be added to the plots?

**Details**

This method is used plot forecasts of GMVAR processes

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

plot.qrtest                      *Quantile residual tests*

---

### Description

quantile\_residual\_tests performs quantile residual tests described by *Kalliovirta and Saikkonen 2010*, testing autocorrelation, conditional heteroskedasticity, and normality.

### Usage

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

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

quantile_residual_tests(
  gmvar,
  lags_ac = c(1:2, 4, 8),
  lags_ch = lags_ac,
  nsimu = 1,
  print_res = TRUE
)
```

### Arguments

x	object of class 'qrtest' generated by the function quantile_residual_tests).
...	currently not used.
digits	the number of decimals to print
gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
lags_ac	a positive integer vector specifying the lags used to test autocorrelation.
lags_ch	a positive integer vector specifying the lags used to test conditional heteroskedasticity.
nsimu	to how many simulations should the covariance matrix Omega used in the qr-tests be based on? If smaller than sample size, then the covariance matrix will be evaluated from the sample. Larger number of simulations might improve the tests size properties but it increases the computation time.
print_res	should the test results be printed while computing the tests?

### Value

Returns an object of class 'qrtest' which has its own print method. The returned object is a list containing the quantile residual test results for normality, autocorrelation, and conditional heteroskedasticity. The autocorrelation and conditional heteroskedasticity results also contain the associated (vectorized) individual statistics divided by their standard errors (see *Kalliovirta and Saikkonen 2010*, s.17-20) under the label \$ind\_stats.

**Methods (by generic)**

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

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [quantile\\_residuals](#), [GIRF](#), [diagnostic\\_plot](#), [predict.gmvar](#), [profile\\_logliks](#), [LR\\_test](#), [Wald\\_test](#), [cond\\_moment\\_plot](#)

**Examples**

```
## These are long running examples that use parallel computing!

# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2) model with default settings
fit12 <- fitGMVAR(data, p=1, M=2)
qrtests12 <- quantile_residual_tests(fit12)
qrtests12
plot(qrtests12)

# Structural GMVAR(1,2) model identified with sign
# constraints. The sign constraints (which fully identify
# the shocks) are in line with the reduced form model,
# so the test results are the same.
W_122 <- matrix(c(1, 1, -1, 1), nrow=2)
fit12s <- fitGMVAR(data, p=1, M=2, structural_pars=list(W=W_122),
  ncalls=10, seeds=1:10)
qrtests12s <- quantile_residual_tests(fit12s)
qrtests12s

# GMVAR(2,2) model with mean parametrization
fit22 <- fitGMVAR(data, p=2, M=2, parametrization="mean",
  ncalls=1, seeds=20)
qrtests22 <- quantile_residual_tests(fit22)
qrtests22
```

```
# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for all regimes
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
fit22c <- fitGMVAR(data, p=2, M=2, constraints=C_mat, ncalls=12)
qrtests22c <- quantile_residual_tests(fit22c, lags_ac=c(1, 4),
                                     nsimu=10000, print_res=TRUE)

qrtests22c
```

---

predict.gmvar

*Predict method for class 'gmvar' objects*

---

## Description

Forecast GMVAR process defined as a class 'gmvar' object. The forecasts are computed by performing independent simulations and using the sample medians or means as point forecasts and empirical quantiles as prediction intervals. For one-step-ahead predictions using the exact conditional mean is also supported.

## Usage

```
## S3 method for class 'gmvar'
predict(
  object,
  ...,
  n_ahead,
  n_simu = 2000,
  pi = c(0.95, 0.8),
  pi_type = c("two-sided", "upper", "lower", "none"),
  pred_type = c("median", "mean", "cond_mean"),
  plot_res = TRUE,
  mix_weights = TRUE,
  nt
)
```

## Arguments

object	an object of class 'gmvar', generated by function fitGMVAR or GMVAR.
...	additional arguments passed to grid (ignored if plot_res==FALSE) which plots grid to the figure.
n_ahead	how many steps ahead should be predicted?
n_simu	to how many independent simulations should the forecast be based on?
pi	a numeric vector specifying the confidence levels of the prediction intervals.
pi_type	should the prediction intervals be "two-sided", "upper", or "lower"?
pred_type	should the prediction be based on sample "median" or "mean"? Or should it be one-step-ahead forecast based on the exact conditional mean ("cond_mean")? Prediction intervals won't be calculated if the exact conditional mean is used.



**plot\_res** should the results be plotted?  
**mix\_weights** TRUE if forecasts for mixing weights should be plotted, FALSE in not.  
**nt** a positive integer specifying the number of observations to be plotted along with the prediction (ignored if plot\_res==FALSE). Default is round(nrow(data)\*0.15).

### Value

Returns a class 'gmvarpred' object containing, among the specifications,...

**\$pred** Point forecasts

**\$pred\_int** Prediction intervals, as [, , d].

**\$mix\_pred** Point forecasts for the mixing weights

**mix\_pred\_int** Individual prediction intervals for mixing weights, as [, , m], m=1,...,M.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

### See Also

[GIRF](#)

### Examples

```
## These are long running examples that use parallel computing!

# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(2,2) model
fit22 <- fitGMVAR(data, p=2, M=2, ncalls=12, seeds=42:53)
p1 <- predict(fit22, n_ahead=20, pred_type="median")
p1
p2 <- predict(fit22, n_ahead=10, nt=20, lty=1)
p2
p3 <- predict(fit22, n_ahead=10, pi=c(0.99, 0.90, 0.80, 0.70),
             nt=30, lty=0)
p3

# GMVAR(1,2) model
fit12 <- fitGMVAR(data, p=1, M=2)
p1 <- predict(fit12, n_ahead=1, pred_type="cond_mean",
             plot_res=FALSE)
p1
```

```

p2 <- predict(fit12, n_ahead=10, pred_type="mean")
p2
p3 <- predict(fit12, n_ahead=10, pi_type="upper")
p3

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(-11.964, 155.024, 11.636, 124.988, 1.314, 0.145, 0.094, 1.292,
-0.389, -0.07, -0.109, -0.281, 1.248, 0.077, -0.04, 1.266, -0.272, -0.074,
0.034, -0.313, 0.903, 0.718, -0.324, 2.079, 7.00, 1.44, 0.742)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(data, p=2, M=2, params=params222s, parametrization="mean",
structural_pars=list(W=W_222))
p1 <- predict(mod222s, n_ahead=10)

```

---

```
print.gmvarpred      Print method for class 'gmvarpred' objects
```

---

## Description

print.gmvarpred is a print method for object generated by predict.gmvar.

## Usage

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

## Arguments

x	object of class 'gmvarpred' generated by predict.gmvar.
...	currently not used.
digits	the number of decimals to print

## Examples

```

# This example uses the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(2,2), d=2 model
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
-0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222, parametrization="mean")
pred222 <- predict(mod222, n_ahead=3, plot_res=FALSE)
print(pred222)
print(pred222, digits=3)

```

---

print.gmvarsum	<i>Summary print method from objects of class 'gmvarsum'</i>
----------------	--

---

### Description

print.gmvarsum is a print method for object 'gmvarsum' generated by summary.gmvar.

### Usage

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

### Arguments

x	object of class 'gmvarsum' generated by summary.gmvar.
...	currently not used.
digits	the number of digits to be printed.

### Examples

```
# This example uses the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(2,2), d=2 model
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
-0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222, parametrization="mean")
sumry222 <- summary(mod222)
print(sumry222)
```

---

print_std_errors	<i>Print standard errors of GMVAR model in the same form as the model estimates are printed</i>
------------------	---

---

### Description

print\_std\_errors prints the approximate standard errors of GMVAR model in the same form as the parameters of objects of class 'gmvar' are printed.

### Usage

```
print_std_errors(gmvar, digits = 3)
```

## Arguments

gmvar            an object of class 'gmvar' created with fitGMVAR or GMVAR.  
 digits            how many digits should be printed?

## Details

The main purpose of print\_std\_errors is to provide a convenient tool to match the standard errors to certain parameter estimates.

Note that if linear constraints are imposed and they involve summations or multiplications, then the AR parameter standard errors are printed separately as they don't correspond one-to-one to the model parameter standard errors.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

## See Also

[profile\\_logliks](#), [fitGMVAR](#), [GMVAR](#), [print.gmvar](#), [swap\\_parametrization](#)

## Examples

```
## These are long running examples that use parallel computing!

# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2) model with default settings
fit12 <- fitGMVAR(data, p=1, M=2)
fit12
print_std_errors(fit12)

# GMVAR(2,2) model with mean parametrization
fit22 <- fitGMVAR(data, p=2, M=2, parametrization="mean")
fit22
print_std_errors(fit22)

# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for all regimes
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
fit22c <- fitGMVAR(data, p=2, M=2, constraints=C_mat)
```

```

fit22c
print_std_errors(fit22c)

# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for all regimes and non-diagonal elements
# the coefficient matrices constrained to zero.
tmp <- matrix(c(1, rep(0, 10), 1, rep(0, 8), 1, rep(0, 10), 1),
  nrow=2*2^2, byrow=FALSE)
C_mat2 <- rbind(tmp, tmp)
fit22c2 <- fitGMVAR(data, p=2, M=2, constraints=C_mat2, ncalls=10)
fit22c2
print_std_errors(fit22c2)

```

---

profile\_logliks

*Plot profile log-likelihoods around the estimates*


---

## Description

profile\_logliks plots profile log-likelihoods around the estimates.

## Usage

```

profile_logliks(
  gmvar,
  which_pars,
  scale = 0.02,
  nrows,
  ncols,
  precision = 200
)

```

## Arguments

**gmvar** an object of class 'gmvar' created with fitGMVAR or GMVAR.

**which\_pars** the profile log-likelihood function of which parameters should be plotted? An integer vector specifying the positions of the parameters in the parameter vector. The parameter vector has the form...

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ . Here  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The default is that profile log-likelihood functions for all parameters are plotted.

scale	a numeric scalar specifying the interval plotted for each estimate: the estimate plus-minus $\text{abs}(\text{scale} \times \text{estimate})$ .
nrows	how many rows should be in the plot-matrix? The default is $\text{max}(\text{ceiling}(\log_2(\text{length}(\text{which\_pars}) - 1), 1)$ .
ncols	how many columns should be in the plot-matrix? The default is $\text{ceiling}(\text{length}(\text{which\_pars}) / \text{nrows})$ . Note that $\text{nrows} \times \text{ncols}$ should not be smaller than the length of <code>which_pars</code> .
precision	at how many points should each profile log-likelihood be evaluated at?

### Details

When the number of parameters is large, it might be better to plot a smaller number of profile log-likelihood functions at a time using the argument `which_pars`.

The red vertical line points the estimate.

### Value

Only plots to a graphical device and doesn't return anything.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

### See Also

[get\\_soc](#), [diagnostic\\_plot](#), [fitGMVAR](#), [GMVAR](#), [GIRF](#), [LR\\_test](#), [Wald\\_test](#), [cond\\_moment\\_plot](#)

### Examples

```
# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form (similar to Kalliovirta et al. 2016).
data(eurusd, package="gmvarkit")
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2) model: 10 estimation rounds with seeds set
# for reproducibility
fit12 <- fitGMVAR(data, p=1, M=2, ncalls=10, seeds=1:10)
fit12
profile_logliks(fit12)
```

```

# Structural GMVAR(1,2) model identified with sign
# constraints.
W_122 <- matrix(c(1, -1, 1, 1), nrow=2)
fit12s <- fitGMVAR(data, p=1, M=2, structural_pars=list(W=W_122),
  ncalls=10, seeds=1:10)
profile_logliks(fit12s)

# GMVAR(2,2) model with mean parametrization
fit22 <- fitGMVAR(data, p=2, M=2, parametrization="mean",
  ncalls=16, seeds=11:26)
profile_logliks(fit22)

# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for both regimes
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
fit22c <- fitGMVAR(data, p=2, M=2, constraints=C_mat)
profile_logliks(fit22c)

# GMVAR(2,2) model with autoregressive parameters restricted
# to be the same for both regimes and non-diagonal elements
# the coefficient matrices constrained to zero.
tmp <- matrix(c(1, rep(0, 10), 1, rep(0, 8), 1, rep(0, 10), 1),
  nrow=2*2^2, byrow=FALSE)
C_mat2 <- rbind(tmp, tmp)
fit22c2 <- fitGMVAR(data, p=2, M=2, constraints=C_mat2)
profile_logliks(fit22c2)

```

---

quantile\_residuals      *Calculate multivariate quantile residuals of a GMVAR model*

---

## Description

quantile\_residuals calculates multivariate quantile residuals (described by *Kalliovirta and Saikkonen 2010*) for a GMVAR model.

## Usage

```
quantile_residuals(gmvar)
```

## Arguments

gmvar                    an object of class 'gmvar' created with fitGMVAR or GMVAR.

## Value

Returns  $((n_{obs} - p) \times d)$  matrix containing the multivariate quantile residuals,  $j$ :th column corresponds to the time series in the  $j$ :th column of the data. The multivariate quantile residuals are

calculated so that the first column quantile residuals are the "unconditioned ones" and the rest condition on all the previous ones in numerical order. Read the cited article by *Kalliovirta and Saikkonen 2010* for details.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

## See Also

[fitGMVAR](#), [GMVAR](#), [quantile\\_residual\\_tests](#), [diagnostic\\_plot](#), [predict.gmvar](#), [profile\\_logliks](#)

## Examples

```
# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(data, p=1, M=2, params=params122)
quantile_residuals(mod122)

# GMVAR(2,2), d=2 model with mean-parametrization:
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
  -0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
  0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
  9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222, parametrization="mean")
quantile_residuals(mod222)

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
  -0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
  0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(data, p=2, M=2, params=params222s, structural_pars=list(W=W_222))
quantile_residuals(mod222s)

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.031, 2.356, 1.786, 3.000, 1.250, 0.060, 0.036,
```



```

1.335, -0.290, -0.083, -0.047, -0.356, 0.934, -0.152, 5.201, 5.883,
3.560, 9.799, 0.368)
mod222c <- GMVAR(data, p=2, M=2, params=params222c, constraints=C_mat)
quantile_residuals(mod222c)

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes and the non-diagonal elements of
# the coefficient matrices constrained to zero.
tmp <- matrix(c(1, rep(0, 10), 1, rep(0, 8), 1, rep(0, 10), 1),
nrow=2*2^2, byrow=FALSE)
C_mat2 <- rbind(tmp, tmp)
params222c2 <- c(0.355, 3.193, -0.114, 2.829, 1.263, 1.338, -0.292,
-0.362, 5.597, 3.456, 9.622, 0.982, -0.327, 5.236, 0.650)
mod222c2 <- GMVAR(data, p=2, M=2, params=params222c2,
constraints=C_mat2)
quantile_residuals(mod222c2)

```

---

quantile\_residuals\_int

*Calculate multivariate quantile residuals of GMVAR model*

---

## Description

quantile\_residuals\_int is a wrapper for quantile\_residuals to compute quantile residuals using parameter values instead of class gmvar object.

## Usage

```

quantile_residuals_int(
  data,
  p,
  M,
  params,
  conditional,
  parametrization,
  constraints = NULL,
  structural_pars = NULL
)

```

## Arguments

data	a matrix or class 'ts' object with $d > 1$ columns. Each column is taken to represent a single time series. NA values are not supported.
p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
params	a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1)x1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m), m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1)x1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2xq)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}, m=1, \dots, M$ .
constraints	a size $(Mpd^2xq)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1), m = 1, \dots, M$ , contains the coefficient matrices and $\psi (qx1)$ contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ $(Mpd^2xpd^2)$ where $I = \text{diag}(p \cdot d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	If NULL a reduced form model is considered. For structural model, should be a list containing the following elements: <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(dxd)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> </ul>

- C\_lambda - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of C\_lambda must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Value

Returns  $((n_{obs} - p) \times d)$  matrix containing the multivariate quantile residuals,  $j$ :th column corresponds to the time series in the  $j$ :th column of the data. The multivariate quantile residuals are calculated so that the first column quantile residuals are the "unconditioned ones" and the rest condition on all the previous ones in numerical order. Read the cited article by *Kalliovirta and Saikkonen 2010* for details.

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

random\_coefmats      *Create random VAR-model ( $d \times d$ ) coefficient matrices A.*

---

### Description

random\_coefmats generates random VAR model coefficient matrices.

### Usage

```
random_coefmats(d, how_many, scale)
```

**Arguments**

d	the number of time series in the system.
how_many	how many ( $d \times d$ ) coefficient matrices $A$ should be drawn?
scale	non-diagonal elements will be drawn from mean zero normal distribution with $sd=0.3/scale$ and diagonal elements from one with $sd=0.6/scale$ . Larger scale will hence more likely result stationary coefficient matrices, but will explore smaller area of the parameter space. Can be for example $1 + \log(2 * \text{mean}(c((p-0.2)^{(1.25)}, d)))$

**Value**

Returns  $((\text{how\_many} * d^2) \times 1)$  vector containing vectorized coefficient matrices  $(\text{vec}(A_1), \dots, \text{vec}(A_{\text{how\_many}}))$ . Note that if  $\text{how\_many} = p$ , then the returned vector equals  $\phi_m$ .

---

random_coefmats2	Create random stationary VAR model ( $d \times d$ ) coefficient matrices $A$ .
------------------	--

---

**Description**

random\_coefmats2 generates random VAR model coefficient matrices.

**Usage**

```
random_coefmats2(p, d, ar_scale = 1)
```

**Arguments**

p	a positive integer specifying the autoregressive order of the model.
d	the number of time series in the system.
ar_scale	a positive real number. Larger values will typically result larger AR coefficients.

**Details**

The coefficient matrices are generated using the algorithm proposed by Ansley and Kohn (1986) which forces stationarity. It's not clear in detail how ar\_scale affects the coefficient matrices. Read the cited article by Ansley and Kohn (1986) and the source code for more information.

**Value**

Returns  $((pd^2) \times 1)$  vector containing stationary vectorized coefficient matrices  $(\text{vec}(A_1), \dots, \text{vec}(A_p))$ .

**References**

- Ansley C.F., Kohn R. 1986. A note on reparameterizing a vector autoregressive moving average model to enforce stationarity. *Journal of statistical computation and simulation*, **24**:2, 99-106.

---

random_covmat	<i>Create random VAR model error term covariance matrix</i>
---------------	---

---

### Description

random\_covmat generates random VAR model ( $dx1$ ) error term covariance matrix  $\Omega$  from (scaled) Wishart distribution for reduced form models and the parameters  $W, \lambda_1, \dots, \lambda_M$  for structural models (from normal distributions).

### Usage

```
random_covmat(d, M, omega_scale, W_scale, lambda_scale, structural_pars = NULL)
```

### Arguments

d	the number of time series in the system.
M	a positive integer specifying the number of mixture components.
omega_scale	a size ( $dx1$ ) strictly positive vector specifying the scale and variability of the random covariance matrices in random mutations. The covariance matrices are drawn from (scaled) Wishart distribution. Expected values of the random covariance matrices are $\text{diag}(\text{omega\_scale})$ . Standard deviations of the diagonal elements are $\sqrt{2/d} * \text{omega\_scale}[i]$ and for non-diagonal elements they are $\sqrt{1/d * \text{omega\_scale}[i] * \text{omega\_scale}[j]}$ . Note that for $d > 4$ this scale may need to be chosen carefully. Default in GAFit is <code>var(stats::ar(data[,i], order.max=10)\$resid</code> . This argument is ignored if structural model is considered.
W_scale	a size ( $dx1$ ) strictly positive vector partly specifying the scale and variability of the random covariance matrices in random mutations. The elements of the matrix $W$ are drawn independently from such normal distributions that the expectation of the main <b>diagonal</b> elements of the first regime's error term covariance matrix $\Omega_1 = WW'$ is $W\_scale$ . The distribution of $\Omega_1$ will be in some sense like a Wishart distribution but with the columns (elements) of $W$ obeying the given constraints. The constraints are accounted for by setting the element to be always zero if it is subject to a zero constraint and for sign constraints the absolute value or negative the absolute value are taken, and then the variances of the elements of $W$ are adjusted accordingly. This argument is ignored if reduced form model is considered.
lambda_scale	a length $M - 1$ vector specifying the <b>standard deviation</b> of the mean zero normal distribution from which the eigenvalue $\lambda_{mi}$ parameters are drawn from in random mutations. As the eigenvalues should always be positive, the absolute value is taken. The elements of <code>lambda_scale</code> should be strictly positive real numbers with the $m - 1$ th element giving the degrees of freedom for the $m$ th regime. The expected value of the main <b>diagonal</b> elements $ij$ of the $m$ th ( $m > 1$ ) error term covariance matrix will be $W\_scale[i] * (d - n\_i)^{-1} * \text{sum}(\text{lambdas} * \text{ind\_fun})$ where the ( $dx1$ ) vector <code>lambdas</code> is drawn from the absolute value of the t-distribution, $n\_i$ is the number of zero constraints in the $i$ th row of $W$ and <code>ind_fun</code> is an indicator function that takes the

value one iff the  $ij$ th element of  $W$  is not constrained to zero. Basically, larger lambdas (or smaller degrees of freedom) imply larger variance.

If the lambda parameters are **constrained** with the  $(d(M-1) \times r)$  constraint matrix  $C_{lambda}$ , then provide a length  $r$  vector specifying the standard deviation of the (absolute value of the) mean zero normal distribution each of the  $\gamma$  parameters are drawn from (the  $\gamma$  is a  $(r \times 1)$  vector). The expected value of the main diagonal elements of the covariance matrices then depend on the constraints.

This argument is ignored if  $M == 1$  or a reduced form model is considered. Default is `rep(3, times=M-1)` if lambdas are not constrained and `rep(3, times=r)` if lambdas are constrained.

As with `omega_scale` and `W_scale`, this argument should be adjusted carefully if specified by hand. **NOTE** that if lambdas are constrained in some other way than restricting some of them to be identical, this parameter should be adjusted accordingly in order to the estimation succeed!

#### structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the  $B$ -matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

#### Value

**For reduced form models:** Returns a  $(d(d+1)/2 \times 1)$  vector containing vech-vectorized covariance matrix  $\Omega$ .

**For structural models:** Returns a length  $d^2 - n_{zeros} - d*(M-1)$  vector of the form  $(Wvec(W), \lambda_2, \dots, \lambda_M)$  where  $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalue parameters of the  $m$ th regime ( $m > 1$ ) and  $n_{zeros}$  is the number of zero constraints in  $W$ . If lambdas are constrained, replace  $d*(M-1)$  in the length with  $r$  and  $\lambda_2, \dots, \lambda_M$  with  $\gamma$ . The operator  $Wvec()$  vectorizes a matrix and removes zeros.

---

random\_ind

*Create random mean-parametrized parameter vector of a GMVAR model that may not be stationary*

---

**Description**

random\_ind generates random mean-parametrized parameter vector that may not be stationary.

**Usage**

```
random_ind(
  p,
  M,
  d,
  constraints = NULL,
  mu_scale,
  mu_scale2,
  omega_scale,
  W_scale,
  lambda_scale,
  structural_pars = NULL
)
```

**Arguments**

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.
constraints	a size $(Mpd^2 \times q)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2 \times 1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi$ ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ ( $Mpd^2 \times pd^2$ ) where $I = diag(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
mu_scale	a size $(dx1)$ vector defining <b>means</b> of the normal distributions from which each mean parameter $\mu_m$ is drawn from in random mutations. Default is colMeans(data). Note that mean-parametrization is always used for optimization in GAfit - even when parametrization=="intercept". However, input (in initpop) and output (return value) parameter vectors can be intercept-parametrized.
mu_scale2	a size $(dx1)$ strictly positive vector defining <b>standard deviations</b> of the normal distributions from which each mean parameter $\mu_m$ is drawn from in random mutations. Default is $2 * sd(data[, i]), i=1, \dots, d$ .
omega_scale	a size $(dx1)$ strictly positive vector specifying the scale and variability of the random covariance matrices in random mutations. The covariance matrices are drawn from (scaled) Wishart distribution. Expected values of the random covariance matrices are $diag(omega\_scale)$ . Standard deviations of the diagonal elements are $\sqrt{2/d} * omega\_scale[i]$ and for non-diagonal elements they are $\sqrt{1/d * omega\_scale[i] * omega\_scale[j]}$ . Note that for $d > 4$ this scale may need to be chosen carefully. Default in GAfit is $var(stats::ar(data[, i], order.max=10)$resid)$ . This argument is ignored if structural model is considered.

- W\_scale** a size  $(dx1)$  strictly positive vector partly specifying the scale and variability of the random covariance matrices in random mutations. The elements of the matrix  $W$  are drawn independently from such normal distributions that the expectation of the main **diagonal** elements of the first regime's error term covariance matrix  $\Omega_1 = WW'$  is  $W\_scale$ . The distribution of  $\Omega_1$  will be in some sense like a Wishart distribution but with the columns (elements) of  $W$  obeying the given constraints. The constraints are accounted for by setting the element to be always zero if it is subject to a zero constraint and for sign constraints the absolute value or negative the absolute value are taken, and then the variances of the elements of  $W$  are adjusted accordingly. This argument is ignored if reduced form model is considered.
- lambda\_scale** a length  $M - 1$  vector specifying the **standard deviation** of the mean zero normal distribution from which the eigenvalue  $\lambda_{mi}$  parameters are drawn from in random mutations. As the eigenvalues should always be positive, the absolute value is taken. The elements of `lambda_scale` should be strictly positive real numbers with the  $m - 1$ th element giving the degrees of freedom for the  $m$ th regime. The expected value of the main **diagonal** elements  $i_j$  of the  $m$ th ( $m > 1$ ) error term covariance matrix will be  $W\_scale[i]*(d - n\_i)^{-1} * \sum(\lambda_{m,i} * ind\_fun)$  where the  $(dx1)$  vector `lambdas` is drawn from the absolute value of the t-distribution,  $n\_i$  is the number of zero constraints in the  $i$ th row of  $W$  and `ind_fun` is an indicator function that takes the value one iff the  $i_j$ th element of  $W$  is not constrained to zero. Basically, larger `lambdas` (or smaller degrees of freedom) imply larger variance.
- If the lambda parameters are **constrained** with the  $(d(M - 1)xr)$  constraint matrix  $C_{lambda}$ , then provide a length  $r$  vector specifying the standard deviation of the (absolute value of the) mean zero normal distribution each of the  $\gamma$  parameters are drawn from (the  $\gamma$  is a  $(rx1)$  vector). The expected value of the main diagonal elements of the covariance matrices then depend on the constraints.
- This argument is ignored if  $M == 1$  or a reduced form model is considered. Default is `rep(3, times=M-1)` if `lambdas` are not constrained and `rep(3, times=r)` if `lambdas` are constrained.
- As with `omega_scale` and `W_scale`, this argument should be adjusted carefully if specified by hand. **NOTE** that if `lambdas` are constrained in some other way than restricting some of them to be identical, this parameter should be adjusted accordingly in order to the estimation succeed!
- structural\_pars** If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:
- `W` - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
  - `C_lambda` - a  $(d(M - 1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of `C_lambda` must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.



See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Value

Returns random mean-parametrized parameter vector that has the same form as the argument `params` in the other functions, for instance, in the function `loglikelihood`.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

random_ind2	<i>Create somewhat random parameter vector of a GMVAR model that is always stationary</i>
-------------	---

---

### Description

`random_ind2` generates random mean-parametrized parameter vector that is always stationary.

### Usage

```
random_ind2(
  p,
  M,
  d,
  mu_scale,
  mu_scale2,
  omega_scale,
  ar_scale = 1,
  W_scale,
  lambda_scale,
  structural_pars = NULL
)
```

### Arguments

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.

- `mu_scale` a size  $(dx1)$  vector defining **means** of the normal distributions from which each mean parameter  $\mu_m$  is drawn from in random mutations. Default is `colMeans(data)`. Note that mean-parametrization is always used for optimization in `GAfit` - even when `parametrization=="intercept"`. However, input (in `initpop`) and output (return value) parameter vectors can be intercept-parametrized.
- `mu_scale2` a size  $(dx1)$  strictly positive vector defining **standard deviations** of the normal distributions from which each mean parameter  $\mu_m$  is drawn from in random mutations. Default is `2*sd(data[,i]), i=1, ..., d`.
- `omega_scale` a size  $(dx1)$  strictly positive vector specifying the scale and variability of the random covariance matrices in random mutations. The covariance matrices are drawn from (scaled) Wishart distribution. Expected values of the random covariance matrices are `diag(omega_scale)`. Standard deviations of the diagonal elements are `sqrt(2/d)*omega_scale[i]` and for non-diagonal elements they are `sqrt(1/d*omega_scale[i]*omega_scale[j])`. Note that for  $d > 4$  this scale may need to be chosen carefully. Default in `GAfit` is `var(stats::ar(data[,i], order.max=10)$resid)`. This argument is ignored if structural model is considered.
- `ar_scale` a positive real number adjusting how large AR parameter values are typically generated in some random mutations. See the function `random_coefmats2` for details. This is ignored when estimating constrained models.
- `W_scale` a size  $(dx1)$  strictly positive vector partly specifying the scale and variability of the random covariance matrices in random mutations. The elements of the matrix  $W$  are drawn independently from such normal distributions that the expectation of the main **diagonal** elements of the first regime's error term covariance matrix  $\Omega_1 = WW'$  is `W_scale`. The distribution of  $\Omega_1$  will be in some sense like a Wishart distribution but with the columns (elements) of  $W$  obeying the given constraints. The constraints are accounted for by setting the element to be always zero if it is subject to a zero constraint and for sign constraints the absolute value or negative the absolute value are taken, and then the variances of the elements of  $W$  are adjusted accordingly. This argument is ignored if reduced form model is considered.
- `lambda_scale` a length  $M - 1$  vector specifying the **standard deviation** of the mean zero normal distribution from which the eigenvalue  $\lambda_{mi}$  parameters are drawn from in random mutations. As the eigenvalues should always be positive, the absolute value is taken. The elements of `lambda_scale` should be strictly positive real numbers with the  $m - 1$ th element giving the degrees of freedom for the  $m$ th regime. The expected value of the main **diagonal** elements  $ij$  of the  $m$ th ( $m > 1$ ) error term covariance matrix will be `W_scale[i]*(d - n_i)^(-1)*sum(lambdas*ind_fun)` where the  $(dx1)$  vector `lambdas` is drawn from the absolute value of the t-distribution, `n_i` is the number of zero constraints in the  $i$ th row of  $W$  and `ind_fun` is an indicator function that takes the value one iff the  $ij$ th element of  $W$  is not constrained to zero. Basically, larger `lambdas` (or smaller degrees of freedom) imply larger variance.
- If the `lambda` parameters are **constrained** with the  $(d(M - 1) \times r)$  constraint matrix  $C_{lambda}$ , then provide a length  $r$  vector specifying the standard deviation of the (absolute value of the) mean zero normal distribution each of the  $\gamma$  parameters are drawn from (the  $\gamma$  is a  $(r \times 1)$  vector). The expected value of the main diagonal elements of the covariance matrices then depend on the constraints.

This argument is ignored if  $M == 1$  or a reduced form model is considered. Default is `rep(3, times=M-1)` if lambdas are not constrained and `rep(3, times=r)` if lambdas are constrained.

As with `omega_scale` and `W_scale`, this argument should be adjusted carefully if specified by hand. **NOTE** that if lambdas are constrained in some other way than restricting some of them to be identical, this parameter should be adjusted accordingly in order to the estimation succeed!

#### structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- `W` - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- `C_lambda` - a  $(d(M-1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of `C_lambda` must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Details

The coefficient matrices are generated using the algorithm proposed by Ansley and Kohn (1986) which forces stationarity. It's not clear in detail how `ar_scale` exactly affects the coefficient matrices but larger `ar_scale` seems to result in larger AR coefficients. Read the cited article by Ansley and Kohn (1986) and the source code for more information.

The covariance matrices are generated from (scaled) Wishart distribution.

Models with AR parameters constrained are not supported!

### Value

Returns random mean-parametrized parameter vector that has the same form as the argument `params` in the other functions, for instance, in the function `loglikelihood`.

### References

- Ansley C.F., Kohn R. 1986. A note on reparameterizing a vector autoregressive moving average model to enforce stationarity. *Journal of statistical computation and simulation*, **24**:2, 99-106.
- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

 reform\_constrained\_pars

*Reform constrained parameter vector into the "standard" form*


---

### Description

reform\_constrained\_pars reforms constrained parameter vector into the form that corresponds to unconstrained parameter vectors.

### Usage

```
reform_constrained_pars(
  p,
  M,
  d,
  params,
  constraints = NULL,
  structural_pars = NULL,
  change_na = FALSE
)
```

### Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1)x1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1)x1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2xq)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma$  ( $rx1$ ) that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1)xr)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization=="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

constraints a size  $(Mpd^2xq)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2x1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi$  ( $qx1$ ) contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2xpd^2$ ) where  $I = \text{diag}(p*d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(dxd)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1)xr)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new ( $rx1$ ) parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

change\_na change NA parameter values of constrained models to -9.999?

### Value

Returns "regular model" parameter vector corresponding to the constraints.

### Warning

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

reform_data	<i>Reform data</i>
-------------	--------------------

---

**Description**

reform\_data reforms the data into a form that is easier to use when calculating log-likelihood values etc.

**Usage**

```
reform_data(data, p)
```

**Arguments**

data	a matrix or class 'ts' object with $d > 1$ columns. Each column is taken to represent a single time series. NA values are not supported.
p	a positive integer specifying the autoregressive order of the model.

**Value**

Returns the data reformed into a  $((n_{obs} - p + 1) \times (dp))$  matrix. The  $i$ :th row of the matrix contains the vector  $(y'_{i-1}, \dots, y'_{i-p})$   $((dp) \times 1)$ , where  $y_i = (y_{1i}, \dots, y_{di})$   $(dx1)$ .

**Warning**

No argument checks!

---

reform_structural_pars	<i>Reform structural parameter vector into the "standard" form</i>
------------------------	--

---

**Description**

reform\_structural\_pars reforms (unconstrained) structural parameter vector into the form that corresponds to reduced form parameter vectors.

**Usage**

```
reform_structural_pars(p, M, d, params, structural_pars = NULL)
```

**Arguments**

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.
params	a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If W is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M - 1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- W - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- C\_lambda - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is

estimated (similarly to AR parameter constraints). The entries of C\_lambda must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

### Details

If the structural parameter vector is a constrained one, use reform\_constrained\_pars first to remove the constraints.

### Value

Returns (unconstrained) "reduced form model" parameter vector.

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

regime_distance	<i>Calculate "distance" between two (scaled) regimes <math>v_m = (\phi_{m,0}, \phi_m, \sigma_m)</math></i>
-----------------	--

---

### Description

regime\_distance calculates "distance" between two scaled regimes. This is used in the genetic algorithm.

### Usage

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

### Arguments

regime\_pars1 a length  $pd^2 + d + d(d+1)/2$  vector  $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$ .  
 regime\_pars2 a length  $pd^2 + d + d(d+1)/2$  vector  $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$ .



**Value**

Returns "distance" between regime\_pars1 and regime\_pars2. Values are scaled before calculating the "distance". Read the source code for more details.

**Warning**

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

reorder_W_columns	<i>Reorder columns of the W-matrix and lambda parameters of a structural GMVAR model.</i>
-------------------	---

---

**Description**

reorder\_W\_columns reorder columns of the W-matrix and lambda parameters of a structural GMVAR model.

**Usage**

```
reorder_W_columns(gmvar, perm)
```

**Arguments**

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
perm	an integer vector of length $d$ specifying the new order of the columns of $W$ . Also lambda parameters of each regime will be reordered accordingly.

**Details**

The order of the columns of  $W$  can be changed without changing the implied reduced form model as long as the order of lambda parameters is also changed accordingly. Note that the constraints imposed on  $W$  (or the B-matrix) will also be modified accordingly.

This function does not support models with constraints imposed on the lambda parameters!

Also all signs in any column of  $W$  can be swapped (without changing the implied reduced form model) with the function swap\_W\_signs but this obviously also swaps the sign constraints in the corresponding columns of  $W$ .

**Value**

Returns an object of class 'gmvar' defining a structural GMVAR model with the modified structural parameters and constraints.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [GIRF](#), [gmvar\\_to\\_sgmvar](#), [swap\\_W\\_signs](#)

**Examples**

```
# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(-11.964, 155.024, 11.636, 124.988, 1.314, 0.145, 0.094, 1.292,
  -0.389, -0.07, -0.109, -0.281, 1.248, 0.077, -0.04, 1.266, -0.272, -0.074,
  0.034, -0.313, 0.903, 0.718, -0.324, 2.079, 7.00, 1.44, 0.742)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(p=2, M=2, d=2, params=params222s, parametrization="mean",
  structural_pars=list(W=W_222))

mod222s

# The same reduced form model, modified W and lambda:
mod222s_2 <- reorder_W_columns(mod222s, perm=2:1)
mod222s_2
```

---

simulateGMVAR

*Simulate from GMVAR process*

---

**Description**

simulateGMVAR simulates observations from a GMVAR process.

**Usage**

```
simulateGMVAR(
  gmvar,
  nsimu,
  init_values = NULL,
  ntimes = 1,
  drop = TRUE,
  seed = NULL,
  girf_pars = NULL
)
```

**Arguments**

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
nsimu	number of observations to be simulated.
init_values	a size $(pxd)$ matrix specifying the initial values to be used in the simulation, where $d$ is the number of time series in the system. The <b>last</b> row will be used as initial values for the first lag, the second last row for second lag etc. If not specified, initial values will be drawn from the stationary distribution of the process.
ntimes	how many sets of simulations should be performed?
drop	if TRUE (default) then the components of the returned list are coerced to lower dimension if $ntimes==1$ , i.e., $\$sample$ and $\$mixing\_weights$ will be matrices, and $\$component$ will be vector.
seed	set seed for the random number generator?
girf_pars	This argument is used internally in the estimation of generalized impulse response functions (see ?GIRF). You should ignore it.

**Details**

The argument `ntimes` is intended for forecasting: a GMVAR process can be forecasted by simulating its possible future values. One can easily perform a large number simulations and calculate the sample quantiles from the simulated values to obtain prediction intervals (see the forecasting example).

**Value**

If `drop==TRUE` and `ntimes==1` (default):  $\$sample$ ,  $\$component$ , and  $\$mixing\_weights$  are matrices. Otherwise, returns a list with...

$\$sample$  a size  $(nsimuxdxntimes)$  array containing the samples: the dimension  $[t, , ]$  is the time index, the dimension  $[ , d, ]$  indicates the marginal time series, and the dimension  $[ , , i]$  indicates the  $i$ :th set of simulations.

$\$component$  a size  $(nsimuxMxntimes)$  matrix containing the information from which mixture component each value was generated from.

$\$mixing\_weights$  a size  $(nsimuxMxntimes)$  array containing the mixing weights corresponding to the sample: the dimension  $[t, , ]$  is the time index, the dimension  $[ , m, ]$  indicates the regime, and the dimension  $[ , , i]$  indicates the  $i$ :th set of simulations.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [diagnostic\\_plot](#), [predict.gmvar](#), [profile\\_logliks](#), [quantile\\_residual\\_tests](#), [GIRF](#)

## Examples

```

# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 process, initial values from the stationary
# distribution
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(p=1, M=2, d=2, params=params122)
set.seed(1)
sim122 <- simulateGMVAR(mod122, nsimu=500)
plot.ts(sim122$sample)
ts.plot(sim122$mixing_weights, col=c("blue", "red"), lty=2)
plot(sim122$component, type="l")

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(-11.964, 155.024, 11.636, 124.988, 1.314, 0.145, 0.094, 1.292,
  -0.389, -0.07, -0.109, -0.281, 1.248, 0.077, -0.04, 1.266, -0.272, -0.074,
  0.034, -0.313, 0.903, 0.718, -0.324, 2.079, 7.00, 1.44, 0.742)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(data, p=2, M=2, params=params222s, parametrization="mean",
  structural_pars=list(W=W_222))
sim222s <- simulateGMVAR(mod222s, nsimu=100)
plot.ts(sim222s$sample)

## FORECASTING EXAMPLE ##
# Forecast 5-steps-ahead, 10000 sets of simulations with initial
# values from the data:
# GMVAR(2,2), d=2 model with mean-parametrization:
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
  -0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
  0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
  9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222, parametrization="mean")
sim222 <- simulateGMVAR(mod222, nsimu=5, ntimes=10000)

# Point forecast + 95% prediction intervals:
apply(sim222$sample, 1:2, quantile, probs=c(0.025, 0.5, 0.972))

# Similar forecast for the mixing weights:
apply(sim222$mixing_weights, 1:2, quantile, probs=c(0.025, 0.5, 0.972))

# GMVAR(2,2), d=2 model with AR parameters restricted to be
# the same for both regimes, custom initial values:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.031, 2.356, 1.786, 3.000, 1.250, 0.060, 0.036,
  1.335, -0.290, -0.083, -0.047, -0.356, 0.934, -0.152, 5.201, 5.883,

```

```

3.560, 9.799, 0.368)
mod222c <- GMVAR(data, p=2, M=2, params=params222c, constraints=C_mat)
sim222c <- simulateGMVAR(mod222c, nsimu=100,
  init_values=matrix(c(30, 30, 80, 80), nrow=2))
plot.ts(sim222c$sample)
ts.plot(sim222c$mixing_weights, col=c("blue", "red"), lty=2)
plot(sim222c$component, type="l")

```

---

smart_covmat	<i>Create random VAR-model (<math>d \times d</math>) error term covariance matrix <math>\Omega</math> fairly close to a given <b>positive definite</b> covariance matrix using (scaled) Wishart distribution</i>
--------------	--

---

### Description

random\_covmat generates random VAR model ( $d \times d$ ) error term covariance matrix  $\Omega$  from (scaled) Wishart distribution that is fairly close to the given matrix.

### Usage

```
smart_covmat(d, M, Omega, W_and_lambdas, accuracy, structural_pars = NULL)
```

### Arguments

- |               |   |
|---------------|---|
| d             | the number of time series in the system.  |
| M             | a positive integer specifying the number of mixture components.   |
| Omega         | a symmetric positive definite ( $d \times d$ ) covariance matrix specifying expected value of the matrix to be generated.   |
| W_and_lambdas | the mean of the normal distribution the new parameters are generated from.<br><b>If lambdas are not constrained:</b> a size $(d^2 - n_{zeros} + d * (M - 1))$ vector $(Wvec(W), \lambda_2, \dots, \lambda_M)$ , where $n_{zeros}$ is the number of zero constraints in $W$ and $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$ .<br><b>If lambdas are constrained:</b> a size $(d^2 - n_{zeros} + r)$ vector $(Wvec(W), \gamma)$ , where $C_\lambda \gamma = (\lambda_2, \dots, \lambda_M)$ , $\gamma$ is of the size $(rx1)$ , and $C_\lambda$ of the size $(d * (M - 1)xr)$ .   |
| accuracy      | a positive real number adjusting how close to the given covariance matrix the returned individual should be.<br>For <b>reduced form models</b> standard deviation of each diagonal element is for reduced form models <ul style="list-style-type: none"> <li>• <math>\omega_{i,i}/accuracy</math> when <math>accuracy &gt; d/2</math></li> <li>• and <math>\sqrt{2/d} * \omega_{i,i}</math> when <math>accuracy \leq d/2</math>.</li> </ul> Wishart distribution is used for reduced form models, but for more details read the source code.<br>For <b>structural models</b> , the parameters are generated from normal distribution with mean given by the argument W_and_lambdas and the standard deviation is $\sqrt{abs(W\_and\_lambdas)/(d + accuracy)}$ . |

**structural\_pars**

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Value**

**For reduced form models:** Returns a  $(d(d+1)/2 \times 1)$  vector containing vech-vectorized covariance matrix  $\Omega$ .

**For structural models:** Returns a length  $d^2 - n\_zeros - d*(M-1)$  vector of the form  $(Wvec(W), \lambda_2, \dots, \lambda_M)$  where  $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalue parameters of the  $m$ th regime ( $m > 1$ ) and  $n\_zeros$  is the number of zero constraints in  $W$ . If lambdas are constrained, replace  $d*(M-1)$  in the length with  $r$  and  $\lambda_2, \dots, \lambda_M$  with  $\gamma$ . The operator  $Wvec()$  vectorizes a matrix and removes zeros.

---

 smart\_ind

*Create random parameter vector of a GMVAR model fairly close to a given parameter vector*

---

**Description**

smart\_ind creates random mean-parametrized parameter vector of a GMVAR model fairly close to a given parameter vector. The result may not be stationary.

**Usage**

```
smart_ind(
  p,
  M,
  d,
  params,
  constraints = NULL,
  accuracy = 1,
  which_random = numeric(0),
  mu_scale,
```

```

mu_scale2,
omega_scale,
ar_scale = 1,
W_scale,
lambda_scale,
structural_pars = NULL
)

```

### Arguments

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If parametrization=="mean", just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

**constraints** a size  $(Mpd^2 \times q)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2 \times 1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi (qx1)$  contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$  ( $Mpd^2 \times pd^2$ ) where  $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

**accuracy** a positive real number adjusting how close to the given parameter vector the returned individual should be. Larger number means larger accuracy. Read the source code for details.

**which\_random** a vector with length between 1 and M specifying the mixture components that should be random instead of close to the given parameter vector. This does not consider constrained AR or lambda parameters.

- `mu_scale` a size  $(dx1)$  vector defining **means** of the normal distributions from which each mean parameter  $\mu_m$  is drawn from in random mutations. Default is `colMeans(data)`. Note that mean-parametrization is always used for optimization in `GAfit` - even when `parametrization=="intercept"`. However, input (in `initpop`) and output (return value) parameter vectors can be intercept-parametrized.
- `mu_scale2` a size  $(dx1)$  strictly positive vector defining **standard deviations** of the normal distributions from which each mean parameter  $\mu_m$  is drawn from in random mutations. Default is `2*sd(data[,i]), i=1, ..., d`.
- `omega_scale` a size  $(dx1)$  strictly positive vector specifying the scale and variability of the random covariance matrices in random mutations. The covariance matrices are drawn from (scaled) Wishart distribution. Expected values of the random covariance matrices are `diag(omega_scale)`. Standard deviations of the diagonal elements are `sqrt(2/d)*omega_scale[i]` and for non-diagonal elements they are `sqrt(1/d*omega_scale[i]*omega_scale[j])`. Note that for  $d > 4$  this scale may need to be chosen carefully. Default in `GAfit` is `var(stats::ar(data[,i], order.max=10)$resid)`. This argument is ignored if structural model is considered.
- `ar_scale` a positive real number adjusting how large AR parameter values are typically generated in some random mutations. See the function `random_coefmats2` for details. This is ignored when estimating constrained models.
- `W_scale` a size  $(dx1)$  strictly positive vector partly specifying the scale and variability of the random covariance matrices in random mutations. The elements of the matrix  $W$  are drawn independently from such normal distributions that the expectation of the main **diagonal** elements of the first regime's error term covariance matrix  $\Omega_1 = WW'$  is `W_scale`. The distribution of  $\Omega_1$  will be in some sense like a Wishart distribution but with the columns (elements) of  $W$  obeying the given constraints. The constraints are accounted for by setting the element to be always zero if it is subject to a zero constraint and for sign constraints the absolute value or negative the absolute value are taken, and then the variances of the elements of  $W$  are adjusted accordingly. This argument is ignored if reduced form model is considered.
- `lambda_scale` a length  $M - 1$  vector specifying the **standard deviation** of the mean zero normal distribution from which the eigenvalue  $\lambda_{mi}$  parameters are drawn from in random mutations. As the eigenvalues should always be positive, the absolute value is taken. The elements of `lambda_scale` should be strictly positive real numbers with the  $m - 1$ th element giving the degrees of freedom for the  $m$ th regime. The expected value of the main **diagonal** elements  $ij$  of the  $m$ th ( $m > 1$ ) error term covariance matrix will be `W_scale[i]*(d - n_i)^(-1)*sum(lambdas*ind_fun)` where the  $(dx1)$  vector `lambdas` is drawn from the absolute value of the t-distribution, `n_i` is the number of zero constraints in the  $i$ th row of  $W$  and `ind_fun` is an indicator function that takes the value one iff the  $ij$ th element of  $W$  is not constrained to zero. Basically, larger `lambdas` (or smaller degrees of freedom) imply larger variance.
- If the `lambda` parameters are **constrained** with the  $(d(M - 1) \times r)$  constraint matrix  $C_{lambda}$ , then provide a length  $r$  vector specifying the standard deviation of the (absolute value of the) mean zero normal distribution each of the  $\gamma$  parameters are drawn from (the  $\gamma$  is a  $(r \times 1)$  vector). The expected value of the main diagonal elements of the covariance matrices then depend on the constraints.



This argument is ignored if  $M == 1$  or a reduced form model is considered. Default is `rep(3, times=M-1)` if lambdas are not constrained and `rep(3, times=r)` if lambdas are constrained.

As with `omega_scale` and `W_scale`, this argument should be adjusted carefully if specified by hand. **NOTE** that if lambdas are constrained in some other way than restricting some of them to be identical, this parameter should be adjusted accordingly in order to the estimation succeed!

#### structural\_pars

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M-1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

#### Value

Returns random mean-parametrized parameter vector that has the same form as the argument `params` in the other functions, for instance, in the function `loglikelihood`.

#### Warning

No argument checks!

#### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

sort_components	<i>Sort components in parameter vector according to mixing weights into a decreasing order</i>
-----------------	--

---

#### Description

`sort_components` sorts mixture components in the parameter vector according to mixing weights into a decreasing order.

**Usage**

```
sort_components(p, M, d, params, structural_pars = NULL)
```

**Arguments**

**p** a positive integer specifying the autoregressive order of the model.  
**M** a positive integer specifying the number of mixture components.  
**d** the number of time series in the system.  
**params** a real valued vector specifying the parameter values.

**For reduced form model:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

If parametrization=="mean", just replace each  $\phi_{m,0}$  with the regimewise mean  $\mu_m$ .  $vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

**structural\_pars**

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- **W** - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- **C\_lambda** - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of **C\_lambda** must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Details**

Constrained parameter vectors are not supported (except for constraints in  $W$ )! For structural models, the order of the first mixture component is fixed by construction, so the rest  $m = 2, \dots, M$  mixture components are rearranged only by the mixing weight parameters.

**Value**

Returns sorted parameter vector...

**For reduced form GMVAR model:** ...with  $\alpha_1 > \dots > \alpha_M$ , that has form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where:

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,1}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For structural GMVAR model:** ...with  $\alpha_2 > \dots > \alpha_M$ , that has form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M, \alpha_1)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ :th coefficient matrix of the  $m$ :th component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ :th component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped.

$\text{vec}()$  is vectorization operator that stack columns of the given matrix into a vector.  $\text{vech}()$  stacks columns of the given matrix from the principal diagonal downwards (including elements on the diagonal) to form a vector. The notation is in line with the cited article by KMS (2016) introducing the GMVAR model.

**Warning**

No argument checks!

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

standard\_errors

*Calculate standard errors for estimates of GMVAR model*

---

**Description**

standard\_errors numerically calculates approximate standard errors for the GMVAR model using square roots of the diagonal of inverse of observed information matrix.

**Usage**

```
standard_errors(
  data,
  p,
  M,
  params,
  conditional = TRUE,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL,
  minval
)
```

**Arguments**

**data** a matrix or class 'ts' object with  $d > 1$  columns. Each column is taken to represent a single time series. NA values are not supported.

**p** a positive integer specifying the autoregressive order of the model.

**M** a positive integer specifying the number of mixture components.

**params** a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d + 1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))$
- and  $\sigma_m = vech(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d + 1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times qx)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, vec(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $vec(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M - 1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If `parametrization="mean"`, just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .

$vec()$  is vectorization operator that stacks columns of a given matrix into a vector.  $vech()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by *Kalliovirta, Meitz and Saikkonen (2016)* introducing the GMVAR model.

conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	"mean" or "intercept" determining whether the model is parametrized with regime means $\mu_m$ or intercept parameters $\phi_{m,0}$ , $m=1,\dots,M$ .
constraints	a size $(Mpd^2 \times q)$ constraint matrix $C$ specifying general linear constraints to the autoregressive parameters. We consider constraints of form $(\phi_1, \dots, \phi_M) = C\psi$ , where $\phi_m = (vec(A_{m,1}), \dots, vec(A_{m,p}))(pd^2 \times 1)$ , $m = 1, \dots, M$ , contains the coefficient matrices and $\psi (qx1)$ contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set $C = [I : \dots : I]'$ $(Mpd^2 \times pd^2)$ where $I = diag(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should <b>not</b> be employed.
structural_pars	<p>If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:</p> <ul style="list-style-type: none"> <li>• <math>W</math> - a <math>(d \times d)</math> matrix with its entries imposing constraints on <math>W</math>: NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.</li> <li>• <math>C\_lambda</math> - a <math>(d(M-1) \times r)</math> constraint matrix that satisfies <math>(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma</math> where <math>\gamma</math> is the new <math>(r \times 1)</math> parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of <math>C\_lambda</math> must be either <b>positive</b> or <b>zero</b>. Ignore (or set to NULL) if the eigenvalues <math>\lambda_{mi}</math> should not be constrained.</li> </ul> <p>See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is <math>W</math> times a time-varying diagonal matrix with positive diagonal entries).</p>
minval	the value that will be returned if the parameter vector does not lie in the parameter space (excluding the identification condition).

### Value

A vector containing the approximate standard errors of the estimates.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

swap\_parametrization    *Swap the parametrization of a GMVAR model*

---

### Description

swap\_parametrization swaps the parametrization of a GMVAR model to "mean" if the current parametrization is "intercept", and vice versa.

### Usage

```
swap_parametrization(gmvar)
```

### Arguments

gmvar                    an object of class 'gmvar' created with fitGMVAR or GMVAR.

### Details

swap\_parametrization is a 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 gmvarKit, the approximate standard errors are only available for parametrized parameters.

### Value

Returns an object of class 'gmvar' defining the specified reduced form or structural GMVAR model. Can be used to work with other functions provided in gmvarKit.

Remark that the first autocovariance/correlation matrix in \$uncond\_moments is for the lag zero, the second one for the lag one, etc.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Kalliovirta L. and Saikkonen P. 2010. Reliable Residuals for Multivariate Nonlinear Time Series Models. *Unpublished Revision of HECER Discussion Paper No. 247*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

### See Also

[fitGMVAR](#), [GMVAR](#), [iterate\\_more](#)

## Examples

```
# These examples use the data 'eurusd' which comes with the
# package, but in a scaled form.
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(data, p=1, M=2, params=params122)
mod122 # intercept parametrization

mod122_2 <- swap_parametrization(mod122)
mod122_2 # mean parametrization

# GMVAR(2,2), d=2 model:
params222 <- c(-11.904, 154.684, 1.314, 0.145, 0.094, 1.292, -0.389,
  -0.070, -0.109, -0.281, 0.920, -0.025, 4.839, 11.633, 124.983, 1.248,
  0.077, -0.040, 1.266, -0.272, -0.074, 0.034, -0.313, 5.855, 3.570,
  9.838, 0.740)
mod222 <- GMVAR(data, p=2, M=2, params=params222, parametrization="mean")
mod222 # mean parametrization

mod222_2 <- swap_parametrization(mod222)
mod222_2 # intercept parametrization

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
  -0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
  0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(data, p=2, M=2, params=params222s, parametrization="intercept",
  structural_pars=list(W=W_222))
mod222s # intercept parametrization

mod222s_2 <- swap_parametrization(mod222s)
mod222s_2 # mean parametrization
```

---

 swap\_W\_signs

*Swap all signs in pointed columns a the  $W$  matrix of a structural GMVAR model.*

---

## Description

swap\_W\_signs swaps all signs in pointed columns a the  $W$  matrix of a structural GMVAR model. Consequently, signs in the columns of the B-matrix are also swapped accordingly.

**Usage**

```
swap_W_signs(gmvar, which_to_swap)
```

**Arguments**

`gmvar` an object of class 'gmvar' created with `fitGMVAR` or `GMVAR`.  
`which_to_swap` a numeric vector of length at most  $d$  and elements in  $1, \dots, d$  specifying the columns of  $W$  whose sign should be swapped.

**Details**

All signs in any column of  $W$  can be swapped without changing the implied reduced form model. Consequently, also the signs in the columns of the B-matrix are swapped. Note that the sign constraints imposed on  $W$  (or the B-matrix) are also swapped in the corresponding columns accordingly.

Also the order of the columns of  $W$  can be changed (without changing the implied reduced form model) as long as the order of lambda parameters is also changed accordingly. This can be done with the function `reorder_W_columns`.

**Value**

Returns an object of class 'gmvar' defining a structural GMVAR model with the modified structural parameters and constraints.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

**See Also**

[fitGMVAR](#), [GMVAR](#), [GIRF](#), [reorder\\_W\\_columns](#), [gmvar\\_to\\_sgmvar](#)

**Examples**

```
# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(-11.964, 155.024, 11.636, 124.988, 1.314, 0.145, 0.094, 1.292,
-0.389, -0.07, -0.109, -0.281, 1.248, 0.077, -0.04, 1.266, -0.272, -0.074,
0.034, -0.313, 0.903, 0.718, -0.324, 2.079, 7.00, 1.44, 0.742)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(p=2, M=2, d=2, params=params222s, parametrization="mean",
structural_pars=list(W=W_222))
mod222s

# The same reduced form model, with signs in the second column of W swapped:
swap_W_signs(mod222s, which_to_swap=2)
```



```
# The same reduced form model, with signs in both column of W swapped:
swap_W_signs(mod222s, which_to_swap=1:2)
```

---

uncond_moments	<i>Calculate the unconditional mean, variance, the first p autocovariances, and the first p autocorrelations of a GMVAR process</i>
----------------	---

---

## Description

uncond\_moments calculates the unconditional mean, variance, the first p autocovariances, and the first p autocorrelations of the given GMVAR process.

## Usage

```
uncond_moments(gmvar)
```

## Arguments

gmvar            an object of class 'gmvar' created with fitGMVAR or GMVAR.

## Details

The unconditional moments are based on the stationary distribution of the process.

## Value

Returns a list with three components:

\$uncond\_mean a length d vector containing the unconditional mean of the process.

\$autocovs an  $(d \times d \times p + 1)$  array containing the lag 0,1,...,p autocovariances of the process. The subset  $[\ , \ j]$  contains the lag j-1 autocovariance matrix (lag zero for the variance).

\$autocors the autocovariance matrices scaled to autocorrelation matrices.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

## See Also

Other moment functions: [cond\\_moments\(\)](#), [get\\_regime\\_autocovs\(\)](#), [get\\_regime\\_means\(\)](#)

**Examples**

```

# GMVAR(1,2), d=2 model:
params122 <- c(0.623, -0.129, 0.959, 0.089, -0.006, 1.006, 1.746,
  0.804, 5.804, 3.245, 7.913, 0.952, -0.037, -0.019, 0.943, 6.926,
  3.982, 12.135, 0.789)
mod122 <- GMVAR(p=1, M=2, d=2, params=params122)
uncond_moments(mod122)

# GMVAR(2,2), d=2 model with AR-parameters restricted to be
# the same for both regimes:
C_mat <- rbind(diag(2*2^2), diag(2*2^2))
params222c <- c(1.031, 2.356, 1.786, 3.000, 1.250, 0.060, 0.036,
  1.335, -0.290, -0.083, -0.047, -0.356, 0.934, -0.152, 5.201, 5.883,
  3.560, 9.799, 0.368)
mod222c <- GMVAR(p=2, M=2, d=2, params=params222c, constraints=C_mat)
uncond_moments(mod222c)

# Structural GMVAR(2, 2), d=2 model identified with sign-constraints:
params222s <- c(1.03, 2.36, 1.79, 3, 1.25, 0.06, 0.04, 1.34, -0.29,
  -0.08, -0.05, -0.36, 1.2, 0.05, 0.05, 1.3, -0.3, -0.1, -0.05, -0.4,
  0.89, 0.72, -0.37, 2.16, 7.16, 1.3, 0.37)
W_222 <- matrix(c(1, 1, -1, 1), nrow=2, byrow=FALSE)
mod222s <- GMVAR(p=2, M=2, d=2, params=params222s, structural_pars=list(W=W_222))
uncond_moments(mod222s)

```

---

uncond_moments_int	<i>Calculate the unconditional mean, variance, the first p autocovariances, and the first p autocorrelations of a GMVAR process</i>
--------------------	---

---

**Description**

uncond\_moments\_int calculates the unconditional mean, variance, the first p autocovariances, and the first p autocorrelations of the specified GMVAR process.

**Usage**

```

uncond_moments_int(
  p,
  M,
  d,
  params,
  parametrization = c("intercept", "mean"),
  constraints = NULL,
  structural_pars = NULL
)

```

**Arguments**

p	a positive integer specifying the autoregressive order of the model.
M	a positive integer specifying the number of mixture components.
d	the number of time series in the system.
params	a real valued vector specifying the parameter values.

**For unconstrained models:** Should be size  $((M(pd^2 + d + d(d+1)/2 + 1) - 1) \times 1)$  and have form  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $v_m = (\phi_{m,0}, \phi_m, \sigma_m)$
- $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))$
- and  $\sigma_m = \text{vech}(\Omega_m)$ ,  $m=1, \dots, M$ .

**For constrained models:** Should be size  $((M(d + d(d+1)/2 + 1) + q - 1) \times 1)$  and have form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \psi, \sigma_1, \dots, \sigma_M, \alpha_1, \dots, \alpha_{M-1})$ , where

- $\psi (qx1)$  satisfies  $(\phi_1, \dots, \phi_M) = C\psi$  where  $C$  is  $(Mpd^2 \times q)$  constraint matrix.

**For structural GMVAR model:** Should have the form  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi_1, \dots, \phi_M, \text{vec}(W), \lambda_2, \dots, \lambda_M)$  where

- $\lambda_m = (\lambda_{m1}, \dots, \lambda_{md})$  contains the eigenvalues of the  $m$ th mixture component.

**If AR parameters are constrained:** Replace  $\phi_1, \dots, \phi_M$  with  $\psi (qx1)$  that satisfies  $(\phi_1, \dots, \phi_M) = C\psi$ , as above.

**If  $W$  is constrained:** Remove the zeros from  $\text{vec}(W)$  and make sure the other entries satisfy the sign constraints.

**If  $\lambda_{mi}$  are constrained:** Replace  $\lambda_2, \dots, \lambda_M$  with  $\gamma (rx1)$  that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $C_\lambda$  is a  $(d(M-1) \times r)$  constraint matrix.

Above,  $\phi_{m,0}$  is the intercept parameter,  $A_{m,i}$  denotes the  $i$ th coefficient matrix of the  $m$ th mixture component,  $\Omega_m$  denotes the error term covariance matrix of the  $m$ th mixture component, and  $\alpha_m$  is the mixing weight parameter. The  $W$  and  $\lambda_{mi}$  are structural parameters replacing the error term covariance matrices (see Virolainen, 2020). If  $M = 1$ ,  $\alpha_m$  and  $\lambda_{mi}$  are dropped. If parametrization="mean", just replace each  $\phi_{m,0}$  with regimewise mean  $\mu_m$ .  $\text{vec}()$  is vectorization operator that stacks columns of a given matrix into a vector.  $\text{vech}()$  stacks columns of a given matrix from the principal diagonal downwards (including elements on the diagonal) into a vector. The notation is in line with the cited article by Kalliovirta, Meitz and Saikkonen (2016) introducing the GMVAR model.

parametrization

"mean" or "intercept" determining whether the model is parametrized with regime means  $\mu_m$  or intercept parameters  $\phi_{m,0}$ ,  $m=1, \dots, M$ .

constraints

a size  $(Mpd^2 \times q)$  constraint matrix  $C$  specifying general linear constraints to the autoregressive parameters. We consider constraints of form  $(\phi_1, \dots, \phi_M) = C\psi$ , where  $\phi_m = (\text{vec}(A_{m,1}), \dots, \text{vec}(A_{m,p}))(pd^2 \times 1)$ ,  $m = 1, \dots, M$ , contains the coefficient matrices and  $\psi (qx1)$  contains the related parameters. For example, to restrict the AR-parameters to be the same for all regimes, set  $C = [I : \dots : I]'$   $(Mpd^2 \times pd^2)$  where  $I = \text{diag}(p \times d^2)$ . Ignore (or set to NULL) if linear constraints should **not** be employed.

**structural\_pars**

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(r \times 1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the B-matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Details**

The unconditional moments are based on the stationary distribution of the process.

**Value**

Returns a list with three components:

`$uncond_mean` a length  $d$  vector containing the unconditional mean of the process.

`$autocovs` an  $(d \times d \times p + 1)$  array containing the lag  $0, 1, \dots, p$  autocovariances of the process. The subset `[, , j]` contains the lag  $j-1$  autocovariance matrix (lag zero for the variance).

`$autocors` the autocovariance matrices scaled to autocorrelation matrices.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Lütkepohl H. 2005. New Introduction to Multiple Time Series Analysis, *Springer*.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

---

unvec

*Reverse vectorization operator*

---

**Description**

unvec forms a square matrix from a vector of stacked columns, stacked by vec.

**Usage**

unvec(d, a)

**Arguments**

- d                    the number of rows in the square matrix to be formed.  
a                    a size  $(d^2 \times 1)$  vector to be unvectorized into a  $(d \times d)$  matrix.

**Value**

a matrix of size  $(d \times d)$ .

**Warning**

No argument checks!

---

unvech	<i>Reverse operator of the parsimonious vectorization operator vech</i>
--------	---

---

**Description**

unvech creates a symmetric matrix from the given vector by copying the lower triangular part to be the upper triangular part as well.

**Usage**

```
unvech(d, a)
```

**Arguments**

- d                    number of rows the square matrix to be formed.  
a                    a size  $(d(d + 1)/2 \times 1)$  vector to be unvectorized into a symmetric  $(d \times d)$  matrix.

**Value**

a symmetric matrix of size  $(d \times d)$ .

**Warning**

No argument checks!

unWvec

*Reverse vectorization operator that restores zeros***Description**

unWvec forms a square matrix from a vector of stacked columns where zeros are removed according to structural parameter constraints.

**Usage**

```
unWvec(Wvector, d, structural_pars = NULL)
```

**Arguments**

**Wvector** a length  $d^2 - n_{zeros}$  vector where  $n_{zeros}$  is the number of zero entries in the matrix  $W$ .

**d** the number of rows in the square matrix to be formed.

**structural\_pars**

If NULL a reduced form model is considered. For structural model, should be a list containing the following elements:

- $W$  - a  $(d \times d)$  matrix with its entries imposing constraints on  $W$ : NA indicating that the element is unconstrained, a positive value indicating strict positive sign constraint, a negative value indicating strict negative sign constraint, and zero indicating that the element is constrained to zero.
- $C\_lambda$  - a  $(d(M - 1) \times r)$  constraint matrix that satisfies  $(\lambda_2, \dots, \lambda_M) = C_\lambda \gamma$  where  $\gamma$  is the new  $(rx1)$  parameter subject to which the model is estimated (similarly to AR parameter constraints). The entries of  $C\_lambda$  must be either **positive** or **zero**. Ignore (or set to NULL) if the eigenvalues  $\lambda_{mi}$  should not be constrained.

See Virolainen (2020) for the conditions required to identify the shocks and for the  $B$ -matrix as well (it is  $W$  times a time-varying diagonal matrix with positive diagonal entries).

**Value**

a  $(d \times d)$  matrix  $W$ .

**Warning**

No argument checks!

---

vec	<i>Vectorization operator</i>
-----	-------------------------------

---

**Description**

vec stacks columns of the given matrix to form a vector.

**Usage**

vec(A)

**Arguments**

A a size ( $d \times d$ ) square matrix to be vectorized.

**Value**

a vector of size ( $d^2 \times 1$ ).

**Warning**

No argument checks!

---

vech	<i>Parsimonious vectorization operator for symmetric matrices</i>
------	---

---

**Description**

vech stacks columns of the given matrix from the principal diagonal downwards (including elements on the diagonal) to form a vector.

**Usage**

vech(A)

**Arguments**

A a size ( $d \times d$ ) symmetric matrix to be vectorized parsimoniously.

**Value**

a vector of size  $(d(d + 1)/2 \times 1)$ .

**Warning**

No argument checks!

---

Wald\_test

*Perform Wald test for a GMVAR or SGMVAR model*


---

### Description

Wald\_test performs a Wald test for a GMVAR or SGMVAR model

### Usage

```
Wald_test(gmvar, A, c, h = 6e-06)
```

```
## S3 method for class 'wald'
print(x, ..., digits = 4)
```

### Arguments

gmvar	an object of class 'gmvar' created with fitGMVAR or GMVAR.
A	a size $(k \times n_{params})$ matrix with full row rank specifying part of the null hypothesis where $n_{params}$ is the number of parameters in the (unconstrained) model. See details for more information.
c	a length $k$ vector specifying part of the null hypothesis. See details for more information.
h	difference used to approximate the derivatives.
x	object of class 'wald' generated by the function Wald_test.
...	other arguments passed to fn
digits	how many significant digits to print?

### Details

Denoting the true parameter value by  $\theta_0$ , we test the null hypothesis  $A\theta_0 = c$ . Under the null, the test statistic is asymptotically  $\chi^2$ -distributed with  $k$  ( $=\text{nrow}(A)$ ) degrees of freedom. The parameter  $\theta_0$  is assumed to have the same form as in the model supplied in the argument gmvar and it is presented in the documentation of the argument params in the function GMVAR (see ?GMVAR).

Finally, note that this function does **not** check whether the specified constraints are feasible (e.g. whether the implied constrained model would be stationary or have positive definite error term covariance matrices).

### Value

Returns an object of class 'wald' containing the test statistic and the related p-value.

### Methods (by generic)

- print: print method



## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2016. Gaussian mixture vector autoregression. *Journal of Econometrics*, **192**, 485-498.
- Virolainen S. 2020. Structural Gaussian mixture vector autoregressive model. Unpublished working paper, available as arXiv:2007.04713.

## See Also

[LR\\_test](#), [fitGMVAR](#), [GMVAR](#), [diagnostic\\_plot](#), [profile\\_logliks](#), [quantile\\_residual\\_tests](#), [cond\\_moment\\_plot](#)

## Examples

```
# Load the data
data(eurusd, package="gmvarkit")
data <- cbind(10*eurusd[,1], 100*eurusd[,2])
colnames(data) <- colnames(eurusd)

# Structural GMVAR(2, 2), d=2 model identified similarly to Cholesky:
W_222 <- matrix(c(1, NA, 0, 1), nrow=2, byrow=FALSE)
fit222s <- fitGMVAR(data, p=2, M=2, structural_pars=list(W=W_222),
                   ncalls=1, seeds=16)

fit222s

# Test whether the lambda parameters (of the second regime) are identical:
# fit222s has parameter vector of length 26 with the lambda parameters
# in elements 24 and 25.
A <- matrix(c(rep(0, times=23), 1, -1, 0), nrow=1, ncol=26)
c <- 0
Wald_test(fit222s, A, c)

# Test whether the off-diagonal elements of the first regime's first
# AR coefficient matrix (A_11) are both zero:
# fit222s has parameter vector of length 26 and the off-diagonal elements
# of the 1st regime's 1st AR coefficient matrix are in the elements 6 and 7.
A <- rbind(c(rep(0, times=5), 1, rep(0, times=20)),
           c(rep(0, times=6), 1, rep(0, times=19)))
c <- c(0, 0)
Wald_test(fit222s, A, c)
```

---

Wvec

*Vectorization operator that removes zeros*

---

## Description

Wvec stacks columns of the given matrix to form a vector and removes elements that are zeros.

**Usage**

Wvec(W)

**Arguments**

W                    a size ( $d \times d$ ) square matrix to be vectorized.

**Value**

a vector of length  $d^2 - n_{zeros}$  where  $n_{zeros}$  is the number of zero entries in the matrix W.

**Warning**

No argument checks!

# Index

- \* **datasets**
  - eurusd, 26
- \* **moment functions**
  - cond\_moments, 19
  - get\_regime\_autocovs, 41
  - get\_regime\_means, 44
  - uncond\_moments, 137
  
- acf, 24
- add\_data, 3, 57
- all\_pos\_ints, 5
- alt\_gmvar, 6
  
- calc\_gradient, 7, 70
- calc\_hessian (calc\_gradient), 7
- change\_parametrization, 9
- change\_regime, 11
- check\_constraints, 13
- check\_data, 14
- check\_gmvar, 15
- check\_null\_data, 15
- check\_parameters, 16
- check\_pMd, 18
- cond\_moment\_plot, 21, 24, 30, 75, 95, 102, 145
- cond\_moments, 19, 41, 44, 137
  
- density, 24
- diag\_Omegas, 24
- diagnostic\_plot, 22, 23, 75, 95, 102, 104, 123, 145
- dlogmultinorm, 25
  
- eurusd, 26
  
- fitGMVAR, 4, 7, 22, 24, 27, 40, 53, 57, 60, 67, 70, 75, 95, 100, 102, 104, 122, 123, 134, 136, 145
- form\_boldA, 32
- format\_valuef, 32
  
- GAFit, 33, 40
- get\_boldA\_eigens, 38
- get\_foc (calc\_gradient), 7
- get\_gradient, 30
- get\_gradient (calc\_gradient), 7
- get\_hessian (calc\_gradient), 7
- get\_IC, 39
- get\_minval, 39
- get\_omega\_eigens, 40
- get\_regime\_autocovs, 21, 41, 44, 137
- get\_regime\_autocovs\_int, 42
- get\_regime\_means, 21, 41, 44, 137
- get\_regime\_means\_int, 45
- get\_soc, 102
- get\_soc (calc\_gradient), 7
- get\_test\_Omega, 47
- get\_unconstrained\_structural\_pars, 50
- GIRF, 30, 51, 57, 60, 95, 97, 102, 122, 123, 136
- GMVAR, 4, 7, 22, 24, 30, 53, 54, 60, 67, 70, 75, 95, 100, 102, 104, 122, 123, 134, 136, 145
- gmvar\_to\_sgmvar, 30, 53, 57, 59, 122, 136
- gmvarkit, 58
  
- in\_paramspace, 60
- in\_paramspace\_int, 62
- is\_stationary, 64
- iterate\_more, 4, 7, 30, 66, 134
  
- logLik.gmvar (GMVAR), 54
- loglikelihood, 68
- loglikelihood\_int, 71
- LR\_test, 22, 24, 30, 53, 74, 95, 102, 145
  
- n\_params, 75
  
- optim, 67
  
- pick\_all\_phi0\_A, 78
- pick\_allA, 76
- pick\_alphas, 80

- pick\_Am, 81
- pick\_Ami, 83
- pick\_lambdas, 85
- pick\_Omegas, 86
- pick\_phi0, 88
- pick\_regime, 89
- pick\_W, 91
- plot.girf (GIRF), 51
- plot.gmvar (GMVAR), 54
- plot.gmvarpred, 93
- plot.qrtest, 94
- predict.gmvar, 24, 30, 53, 95, 96, 104, 123
- print.girf (GIRF), 51
- print.gmvar, 100
- print.gmvar (GMVAR), 54
- print.gmvarpred, 98
- print.gmvarsum, 99
- print.lr (LR\_test), 74
- print.qrtest (plot.qrtest), 94
- print.wald (Wald\_test), 144
- print\_std\_errors, 30, 99
- profile\_logliks, 8, 22, 24, 30, 53, 67, 75, 95, 100, 101, 104, 123, 145
  
- quantile\_residual\_tests, 22, 24, 30, 53, 75, 104, 123, 145
- quantile\_residual\_tests (plot.qrtest), 94
- quantile\_residuais, 95, 103
- quantile\_residuais\_int, 105
  
- random\_coefmats, 107
- random\_coefmats2, 108
- random\_covmat, 109
- random\_ind, 110
- random\_ind2, 113
- reform\_constrained\_pars, 116
- reform\_data, 118
- reform\_structural\_pars, 118
- regime\_distance, 120
- reorder\_W\_columns, 30, 53, 57, 60, 121, 136
- residuals.gmvar (GMVAR), 54
  
- simulateGMVAR, 30, 53, 122
- smart\_covmat, 125
- smart\_ind, 126
- sort\_components, 129
- standard\_errors, 131
- summary.gmvar (GMVAR), 54
  
- swap\_parametrization, 30, 57, 100, 134
- swap\_W\_signs, 30, 53, 57, 60, 122, 135
  
- uncond\_moments, 21, 41, 44, 137
- uncond\_moments\_int, 138
- unvec, 140
- unvech, 141
- unWvec, 142
  
- vec, 143
- vech, 143
  
- Wald\_test, 22, 24, 30, 53, 75, 95, 102, 144
- Wvec, 145