Package 'micvar'

December 8, 2025

Type Package

Title Order Selection in Vector Autoregression by Mean Square Information Criteria

Version 0.1.0

Description Implements order selection for Vector Autoregressive (VAR) models using the Mean Square Information Criterion (MIC). Unlike standard methods such as AIC and BIC, MIC is likelihood-free. This method consistently estimates VAR order and has robust performance under model misspecification. For more details, see Hellstern and Shojaie (2025) <doi:10.48550/arXiv.2511.19761>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Imports MASS, matrixcalc, Rdpack, stats

RdMacros Rdpack

NeedsCompilation no

Author Michael Hellstern [aut, cre]

Maintainer Michael Hellstern <mikeh1@uw.edu>

Repository CRAN

Date/Publication 2025-12-08 08:10:10 UTC

Contents

gen_coef_mat *Simulate coefficient matrices with specified density*

Description

Simulates coefficient matrices used to generate data from a vector autoregressive process.

Usage

```
gen_coef_mat(k, coefmin, coefmax, dens)
```

Arguments

k Integer. Dimension of process.

coefmin Numeric. Minimum value of coefficient. See Details.

coefmax Numeric. Maximum value of coefficient. See Details.

dens Numeric. Must be between 0 and 1. Specifies the proportion of non-zero entries in the coefficient matrix. The number of non-zero entries is computed as floor(k^2*dens).

Details

Coefficient values are drawn from a Uniform(coefmin, coefmax) or a Uniform(-coefmax, -coefmin) each with 50% probability.

Value

k x k matrix.

Examples

```
# bivariate coefficient matrix
coefmat <- gen_coef_mat(k = 2, coefmin = 0.1, coefmax = 0.3, dens = 0.8)
print(coefmat)
```

micvar *Estimate order by mean square information criteria (MIC)*

Description

Fits an autoregressive model to the data where the order is selected by minimizing the mean square information criteria. Model fitting is performed using ar. Any of the methods available in the method argument of ar can be used.

Usage

```
micvar(
  x,
  pmax,
  pmaxst = 2 * pmax,
  method = "ols",
  na.action = stats::na.fail,
  series = deparse1(substitute(x)),
  demean = TRUE,
  ...
)
```

Arguments

`x` n x p time series data matrix. Can be univariate or multivariate time series. If x is not a matrix it will be coerced using `as.matrix(x)`.

`pmax` Integer. Maxmium number of lags to consider. Considered lags will to be `0`,`1`,...,`pmax`.

`pmaxst` Integer (default is `2pmax`). Maximum lag used for computing self-tuned lambda. Must be larger than `pmax`.

`method` Character string (default is "ols"). Specifies method to fit the model. Options are: `c("ols", "burg", "mle", "yule-walker", "yw")`. Note this function uses ar to perform model fitting.

`na.action` Function for missing values (default is `na.fail`). See the `na.action` argument in ar.

`series` Character string. Name of series. See the `series` argument in ar.

`demean` Boolean (default is TRUE). Whether or not to demean the series. See the `demean` argument in ar.

`...` Additional arguments for specific method. See ar and its various methods such as ar.yw and ar.ols and their corresponding arguments.

Details

This function uses the ar functions for fitting. For relevant details of those methods see the Details section of ar.

Value

List with elements. Many of these elements are similar to ar.

`order` Order of fitted model selected by MIC

`penalized_losses`
Numeric vector of penalized losses for orders `0`,`1`,...,`pmax`.

`ar` Estimated autoregression coefficients. See the `ar` return value from ar.

`var.pred` Prediction variance. See the `var.pred` return value from ar.

`x.mean` Estimated mean. See the `x.mean` return value from ar.

`x.intercept` Intercept. See the `x.intercept` return value from ar.

`n.used` Number of observations in the time series including missing. See the `n.used` return value from ar.

`n.obs` Number of non-missing observations. See the `n.obs` return value from ar.

`pmax` The value of `pmax` argument.

`partialacf` Estimate of partial autocorrelation. See the `partialacf` return value from ar.

`resid` Residuals from fitted model. See the `resid` return value from ar.

`method` Value of `method` argument.

`series` Name of the series. See the `series` return value from ar.

`call` Function call.

`asy.var.coef` Asymptotic-theory variance matrix of coefficient estimates. See the `asy.var.coef` return value from ar.

Examples

```
# multivariate example - default is OLS
VAR3_2_A <- list(gen_coef_mat(3, 0.1, 0.3, 0.8), # lag 1
                 gen_coef_mat(3, 0.1, 0.4 , 0.5)) # lag 2
x <- sim_var(VAR3_2_A, n = 5000)
mic_model <- micvar(x, pmax = 10)

# burg and yule-walker examples
mic_model_burg <- micvar(x, pmax = 10, method = ”burg”)
mic_model_yw <- micvar(x, pmax = 10, method = ”yw”)

# univariate example
ar_coefs <- list(matrix(0.3,nrow=1), matrix(0.1,nrow=1))
x <- sim_var(ar_coefs, n = 5000)
mic_model <- micvar(x, pmax = 10)
```

sim_var *Simulate data from a vector autoregressive model with specified coefficient matrices*

Description

Simulates data from a stable vector autoregressive model with Gaussian innovations and specified coefficient matrices. Stability of the process is verified using verify_stability.

Usage

```
sim_var(A, n, mu = NULL, Sigma = NULL, burn_in = 500)
```

Arguments

`A` List of coefficient matrices. Each element in A must be a square matrix. Dimension of matrix determines the number of variables. Length of A determines the order of the process. In the case of univariate time series each entry of A should be a 1 x 1 matrix.

`n` Integer. Number of data points to simulate.

`mu` Vector (default 0s). Means of Gaussian innovations.

`Sigma` Square matrix (default Identity). Variance of Gaussian innovations.

`burn_in` Integer (default 500). Number of observations used to start up simulated process. In total `n` + `burn_in` observations are simulated but the first `burn_in` are discarded.

Value

`n` x `k` data matrix.

Examples

```
# multivariate
VAR3_2_A <- list(gen_coef_mat(3, 0.1, 0.3, 0.8), # lag 1
                 gen_coef_mat(3, 0.1, 0.4 , 0.5)) # lag 2
x <- sim_var(VAR3_2_A, n = 1000)

# univariate
AR2 <- list(matrix(0.5), matrix(0.2))
x <- sim_var(AR2, n = 1000)

# non-identity covariance of Gaussian innovations
Sigma <- matrix(c(1,0.5,0.9,0.5,1.5,0.7,0.9,0.7,1.25), nrow = 3)
x <- sim_var(VAR3_2_A, n = 1000, Sigma = Sigma)
```

`verify_stability` *Verify stability of a vector autoregressive model*

Description

Stability is verified using the method the method on pages 14-17 of (Lütkepohl 2005). Specifically we generate the coefficient matrix for the VAR(1) representation of the process and check that all eigenvalues have modulus less than 1.

Usage

```
verify_stability(A)
```

Arguments

`A` List of coefficient matrices.

Value

None. Throws error if not stable process.

References

Lütkepohl H (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.

Examples

```
VAR3_2_A <- list(gen_coef_mat(3, 0.1, 0.3, 0.8), # lag 1
                 gen_coef_mat(3, 0.1, 0.4 , 0.5)) # lag 2
verify_stability(VAR3_2_A)
```

Index