Package ‘VARshrink’

October 9, 2019

Title  Shrinkage Estimation Methods for Vector Autoregressive Models

Version 0.3.1

Description  Vector autoregressive (VAR) model is a fundamental and effective approach for multivariate time series analysis. Shrinkage estimation methods can be applied to high-dimensional VAR models with dimensionality greater than the number of observations, contrary to the standard ordinary least squares method. This package is an integrative package delivering nonparametric, parametric, and semiparametric methods in a unified and consistent manner, such as the multivariate ridge regression in Golub, Heath, and Wahba (1979) <doi:10.2307/1268518>, a James-Stein type nonparametric shrinkage method in Opgen-Rhein and Strimmer (2007) <doi:10.1186/1471-2105-8-S2-S3>, and Bayesian estimation methods using noninformative and informative priors in Lee, Choi, and S.-H. Kim (2016) <doi:10.1016/j.csda.2016.03.007> and Ni and Sun (2005) <doi:10.1198/073500104000000622>.

License  GPL-3

Depends  R (>= 3.5.0)

Imports  vars (>= 1.5.3), ars (>= 0.6), corpcor (>= 1.6.9), strucchange, stats, MASS, mvtnorm

Suggests  knitr, rmarkdown, rticles, kableExtra

URL  https://github.com/namgillee/VARshrink/

BugReports  https://github.com/namgillee/VARshrink/issues/

Encoding  UTF-8

LazyData  true

RoxygenNote  6.1.1

VignetteBuilder  knitr

NeedsCompilation  no

Author  Namgil Lee [aut, cre] (<https://orcid.org/0000-0003-0593-9028>), Heon Young Yang [ctb], Sung-Ho Kim [aut]

Maintainer  Namgil Lee <namgil.lee@kangwon.ac.kr>

Repository  CRAN

Date/Publication  2019-10-09 15:10:03 UTC
Description

Returns the estimated coefficient matrices of the lagged endogenous variables of a VAR(p) model. This is a modification of vars::Acoef() for the class "varshrinkset".

Usage

Acoef_sh(x)

Arguments

x An object of class "varshrinkeset", generated by VARshrink().
Details

Consider VAR(p) model:

\[ y_t = A_1 y_{t-1} + ... + A_p y_{t-p} + C d_t + e_t. \]

The function returns the K-by-K matrices \( A_1, ..., A_p \) as a list object.

Value

A list object with K-by-K VAR coefficient matrices \( A_1, A_2, ..., A_p \)

See Also

\( \text{Acoef} \)

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
Acoef_sh(estim)

Description

Performs univariate and multivariate ARCH-LM tests for a VAR. This is a modification of \( \text{vars::arch.test()} \) for the class "varshrinkest".

Usage

\[
\text{arch.test.sh}(x, \text{lags.single} = 16, \text{lags.multi} = 5, \text{multivariate.only} = \text{TRUE})
\]

Arguments

- \( x \): An object of class "varshrinkest" obtained by \( \text{VARshrink()} \)
- \( \text{lags.single} \): An integer of the lag order used for univariate ARCH statistics.
- \( \text{lags.multi} \): An integer of the lag order used for multivariate ARCH statistic.
- \( \text{multivariate.only} \): If TRUE, only the multivariate statistic is computed.

See Also

\( \text{arch.test} \)
Examples

```r
data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
arch.test_sh(estim)
```

---

**Bcoef_sh**

*Coefficient matrix*

---

**Description**

Returns the estimated coefficients of a VAR(p) model as a matrix. This is a modification of `vars::Bcoef()` for the class "varshrinkest".

**Usage**

```
Bcoef_sh(x)
```

**Arguments**

- `x` An object of class "varshrinkest" generated by `VARshrink()`.

**Details**

Consider VAR(p) model:

\[ y_t = A_1 y_{t-1} + ... + A_p y_{t-p} + C d_t + e_t. \]

The function returns the concatenated matrix \((A_1, ..., A_p, C)\) as a matrix object.

**Value**

A matrix holding the estimated coefficients of a VAR.

**See Also**

`Bcoef`

**Examples**

```r
data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
Bcoef_sh(estim)
```
**BQ_sh**  
*BQ function for class "varshrinkest"

**Description**
This is a modification of vars::BQ() for the class "varshrinkest".

**Usage**

```r
BQ_sh(x)
```

**Arguments**

*x*
An object of class "varshrinkest" obtained by VARshrink().

**See Also**

*BQ*

**Examples**

```r
data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
BQ_sh(estim)
```

---

**calcSSE_Acoef**  
*Sum of squared errors (SSE) between coefficients of two VARs*

**Description**
Compute sum of squared errors of coefficients of lagged endogenous variables (Acoef) of two VAR models.

**Usage**

```r
calcSSE_Acoef(Acoef1, Acoef2)
```

**Arguments**

*Acoef1, Acoef2*
Each one is a list object with K-by-K coefficient matrices of lagged endogenous variables. See help(Acoef_sh), or, help(Acoef).
Details

Consider VAR(p) model:
\[ y_t = A_1 y_{t-1} + \ldots + A_p y_{t-p} + C d_t + e_t. \]

The SSE of two VAR(p) models is expressed as
\[ \sum_{k=1}^{p} \sum_{i=1}^{K} \sum_{j=1}^{K} (A_{ki} - (A'_{ki}))^2. \]

Value

SSE value.

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim1 <- VARshrink(y, p = 2, type = "const", method = "fbayes")
Acoef1 <- Acoef_sh(estim1)
estim2 <- VARshrink(y, p = 2, type = "const", method = "ridge")
Acoef2 <- Acoef_sh(estim2)
calcSSE_Acoef(Acoef1, Acoef2)

causality_sh Causality Analysis for class "varshrinkest"

Description

A modification of vars::causality() for the class "varshrinkest".

Usage

causality_sh(x, cause = NULL, vcov. = NULL, boot = FALSE, 
boot.runs = 100)

Arguments

x An object of class "varshrinkest" obtained by VARshrink().
cause, vcov., boot, boot.runs
Other arguments for causality analysis; see help(causality) for details.

See Also

causality

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
causality_sh(estim, cause = "e")
**convPsi2varresult**

Convert format for VAR coefficients from Psi to varresult

**Description**

Convert a matrix of VAR coefficients estimated by a shrinkage method into a list of "shrinklm" object, where the class "shrinklm" inherits the class "lm".

**Usage**

```r
convPsi2varresult(Psi, Y, X, lambda0, type = c("const", "trend", "both", "none"), ybar = NULL, xbar = NULL, Q_values = NULL, callstr = "")
```

**Arguments**

- `Psi`: An M-by-K matrix of VAR coefficients
- `Y`: An N-by-K data matrix of dependent variables
- `X`: An N-by-M data matrix of regressors
- `lambda0`: A rescaled shrinkage intensity parameter, based on which the effective number of parameters is computed by
  \[
  \text{Trace}(X'X + \lambda_0 I)^{-1}X'
  \]
- `type`: Type of deterministic variables in the VAR estimation problem. Either of "const", "trend", "both", or "none".
- `ybar, xbar`: NULL if Y and X are not centered. Mean vectors if Y and X had been centered. If Y and X had been centered (ybar and xbar are not NULL) and type is "const" or "both", then the coefficients for the constant term is computed and concatenated to the coefficients.
- `Q_values`: Nonnegative weight vector of length N. Default is NULL. Take weights on rows (samples) of Y and X by \(\sqrt{Q}\).
- `callstr`: The call to VARshrink().

**Details**

Consider VAR(p) model:

\[
y_t = A_1y_{t-1} + \ldots + A_py_{t-p} + Cd_t + e_t.
\]

It can be written in the matrix form:

\[
Y = XPsi + E,
\]

where Psi is a concatenated M-by-K matrix, Psi = (A_1, ..., A_p, C)^T. It can be written in the multiple linear regression form of a VAR(p) model:

\[
y_j = Xpsi_j + e_j, \quad j = 1, ..., K,
\]
where \( y_j, \psi_j, \) and \( e_j \) are the \( j \)-th column vectors of \( Y, \Psi, \) and \( E, \) respectively. This function converts \( \Psi \) into a list of "shrinklm" objects, where each "shrinklm" object contains the length-\( M \) vector \( \psi_j \) as coefficients.

Considering that each coefficient vector \( \psi_j \) is estimated by a shrinkage method, the effective number of parameters, \( k_{eff} \), is computed as:

\[
k_{eff} = \text{trace}(X(X^TX + \lambda_0 * I)^{-1}X^T).
\]

Then, the degree of freedom of residuals is computed as:

\[
df\text{.residual} = N - k_{eff},
\]

where \( N \) is the number of rows of data matrices \( Y \) and \( X \).

Value

A list object with objects of class c("shrinklm", "lm"). Each "shrinklm" object has components: coefficients, residuals, fitted.values, rank, df.residual, lambda0, call, terms, svd

---

createVARCoefs_ltriangular

*Create coefficients of a VAR model*

**Description**

Randomly create sparse lower-triangular matrices for VAR coefficients of lagged endogenous variables, and set a constant vector.

**Usage**

createVARCoefs_ltriangular(p = 1, K = 5, diag_val = 1/p, num_nonzero = 0, const_vector = NULL, range_min = 0.2, range_max = 1/p)

**Arguments**

- \( p \) : lag order
- \( K \) : Number of time series variables.
- diag_val : diagonal values of \( A_1, ..., A_p \)
- num_nonzero : Number of nonzero entries on the lower-triangular parts of \( A_1, ..., A_p \)
- const_vector : constant vector \( c \) of the VAR model
- range_min, range_max : Each nonzero off-diagonal entry of coefficient matrices is drawn uniformly from the interval \([-\text{range.max}, -\text{range.min}] \cup [\text{range.min}, \text{range.max}]\)
Details

Consider VAR(p) model:

\[ y_t = A_1 y_{t-1} + \ldots + A_p y_{t-p} + c + e_t, \]

with the constant deterministic variable \( d_t = 1 \). The function creates the coefficient matrices \( A_1, \ldots, A_p \) and constant vector \( c \).

Diagonal elements of each K-by-K matrix \( A_k \) are all equal to diag_val, and off-diagonal elements are all zero except for a few randomly selected nonzero elements. Nonzero off-diagonal elements are selected from lower-triangular parts of \( A_i \) and the values are drawn from a uniform distribution over \([-\text{range}\_\text{max}, -\text{range}\_\text{min}] \cup [\text{range}\_\text{min}, \text{range}\_\text{max}]\).

Value

A list object with components $A$ and $c$. $A$ is a list of K-by-K matrices $A_1, \ldots, A_p$, and $c$ is a constant vector of length K.

Examples

```r
p <- 1; K <- 20;
const_vector <- c(rep(0.2, 5), rep(0.7, 15))
createVARCoefs_ltriangular(p = p, K = K, diag_val = 0.6,
num_nonzero = K, const_vector = const_vector, range_max = 1)
```

Description

Computes the forecast error variance decomposition of a VAR(p) for n.ahead steps. This is a modification of vars::fevd() for the class "varshrinkest".

Usage

```r
## S3 method for class 'varshrinkest'
fevd(x, n.ahead = 10, ...)
```

Arguments

- `x` Object of class 'varshrinkest'; generated by VARshrink().
- `n.ahead` Integer specifying the steps.
- `...` Currently not used.

See Also

`fevd`
**irf.varshrinkest**  
*Impulse response function*

**Description**

Computes the impulse response coefficients of a VAR(p) (or transformed VECM to VAR(p)) for \( n \text{.ahead} \) steps. This is a modification of `vars::irf()` for the class "varshrinkest".

**Usage**

```r
## S3 method for class 'varshrinkest'
irf(x, impulse = NULL, response = NULL,
    n.ahead = 10, ortho = TRUE, cumulative = FALSE, boot = TRUE,
    ci = 0.95, runs = 100, seed = NULL, ...)
```

**Arguments**

- `x` Object of class 'varshrinkest'; generated by `VARshrink()`
- `impulse` A character vector of the impulses, default is all variables.
- `response` A character vector of the responses, default is all variables.
- `n.ahead` Integer specifying the steps.
- `ortho` Logical, if TRUE (the default) the orthogonalised impulse response coefficients are computed (only for objects of class 'varshrinkest').
- `cumulative` Logical, if TRUE the cumulated impulse response coefficients are computed. The default value is false.
- `boot` Logical, if TRUE (the default) bootstrapped error bands for the impulse response coefficients are computed.
- `ci` Numeric, the confidence interval for the bootstrapped errors bands.
- `runs` An integer, specifying the runs for the bootstrap.
- `seed` An integer, specifying the seed for the rng of the bootstrap.
- `...` Currently not used.

**See Also**

- `irf`
Description

Estimate regression coefficients and scale matrix for noise by using Gibbs MCMC algorithm. The function assumes 1) multivariate t-distribution for noise as a sampling distribution, and 2) noninformative priors for regression coefficients and scale matrix for noise.

Usage

```r
lm_full_Bayes_SR(Y, X, dof = Inf, burnincycle = 1000,
                 mcmccycle = 2000)
```

Arguments

- **Y**: An N x K matrix of dependent variables.
- **X**: An N x M matrix of regressors.
- **dof**: Degree of freedom for multivariate t-distribution. If `dof = Inf` (default), then multivariate normal distribution is applied and weight vector `q` is not estimated. If `dof = NULL` or `dof <= 0`, then `dof` and `q` are estimated automatically. If `dof` is a positive number, `q` is estimated.
- **burnincycle, mcmccycle**: Number of burnin cycles is the number of initially generated sample values to drop. Number of MCMC cycles is the number of generated sample values to compute estimates.

Details

Consider the multivariate regression:

\[ Y = X \Psi + e, \quad e \sim \text{mvt}(0, \text{dof}, \Sigma). \]

\( \Psi \) is a M-by-K matrix of regression coefficients and \( \Sigma \) is a K-by-K scale matrix for multivariate t-distribution for noise.

Sampling distribution for noise \( e \) is multivariate t-distribution with degree of freedom `dof` and scale matrix \( \Sigma \): \( e \sim \text{mvt}(0, \text{dof}, \Sigma) \). The priors are noninformative priors: 1) the shrinkage prior for regression coefficients \( \Psi \), and 2) the reference prior for scale matrix \( \Sigma \).

The function implements Gibbs MCMC algorithm for estimating regression coefficients \( \Psi \) and scale matrix \( \Sigma \).

Value

A list object with estimated parameters: \( \Psi \), \( \Sigma \), `dof`, `delta` (delta is the reciprocal of lambda), and lambda. Additional components are `se.param` (standard error of the parameters) and LINEX-VARmodel (estimates under LINEX loss).
References


---

**lm_multiv_ridge**

*Multivariate Ridge Regression*

**Description**

Estimate regression coefficients by using ridge regression.

**Usage**

```r
lm_multiv_ridge(Y, X, lambda = 0, do_scale = FALSE)
```

**Arguments**

- `Y`: An N x K matrix of dependent variables.
- `X`: An N x M matrix of regressors.
- `lambda`: Numeric vector of lambda values
- `do_scale`: If true, X is centered and scaled, and Y is centered.

**Details**

Consider the multivariate regression:

\[ Y = X \Psi + \epsilon. \]

\( \Psi \) is a M-by-K matrix of regression coefficients. The ridge regression estimate for the coefficients is

\[ \Psi = (X'X + \lambda I)^{-1} X'Y. \]

**Value**

A list object with the components: 1) `Psi` - A list of estimated Psi matrices, 2) `lambda` - A vector of lambda values, 3) `GCV` - A vector of GCV values

**References**

Semiparametric Bayesian Shrinkage Estimation Method for Multivariate Regression

Description

Estimate regression coefficients and scale matrix for noise by using a parameterized cross validation (PCV). The function assumes 1) multivariate t-distribution for noise as a sampling distribution, and 2) informative priors for regression coefficients and scale matrix for noise.

Usage

```r
lm_semi_Bayes_PCV(Y, X, dof = Inf, lambda = NULL, lambda_var = NULL,
prior_type = c("NCJ", "CJ"), num_folds = 5, m0 = ncol(Y))
```

Arguments

- **Y**: An N x K matrix of dependent variables.
- **X**: An N x M matrix of regressors.
- **dof**: Degree of freedom for multivariate t-distribution. If dof = Inf (default), then multivariate normal distribution is applied and weight vector q is not estimated. If dof = NULL or a numeric vector, then dof is selected by K-fold CV automatically and q is estimated.
- **lambda**: If NULL or a vector of length >= 2, it is selected by PCV.
- **lambda_var**: If NULL, it is selected by a Stein-type shrinkage method.
- **prior_type**: "NCJ" for non-conjugate prior and "CJ" for conjugate prior for scale matrix Sigma.
- **num_folds**: Number of folds for PCV.
- **m0**: A hyperparameter for inverse Wishart distribution for Sigma

Details

Consider the multivariate regression:

\[ Y = X \Psi + \varepsilon, \quad \varepsilon \sim \text{mvt}(0, \text{dof}, \Sigma). \]

Psi is a M-by-K matrix of regression coefficients and Sigma is a K-by-K scale matrix for multivariate t-distribution for noise.

Sampling distribution for noise \( \varepsilon \) is the multivariate t-distribution with degree of freedom \( \text{dof} \) and scale matrix Sigma: \( \varepsilon \sim \text{mvt}(0, \text{dof}, \Sigma) \). The priors are informative priors: 1) a shrinkage prior for regression coefficients Psi, and 2) inverse Wishart prior for scale matrix Sigma, which can be either non-conjugate ("NCJ") or conjugate ("CJ") to the shrinkage prior for coefficients Psi.

The function implements parameterized cross validation (PCV) for selecting a shrinkage parameter lambda for estimating regression coefficients \( 0 < \lambda \leq 1 \). In addition, the function uses a Stein-type shrinkage method for selecting a shrinkage parameter lambda_var for estimating variances of time series variables.
References


Description

Estimate regression coefficients and scale matrix for noise by using semiparametric Bayesian shrinkage estimator, whose shrinkage parameters are selected by K-fold cross validation (KCV).

Usage

`lm_ShVAR_KCV(Y, X, dof = Inf, lambda = NULL, lambda_var = NULL, prior_type = c("NCJ", "CJ"), num_folds = 5, m0 = ncol(Y))`

Arguments

- `Y` An N x K matrix of dependent variables.
- `X` An N x M matrix of regressors.
- `dof` Degree of freedom for multivariate t-distribution. If dof = Inf (default), then multivariate normal distribution is applied and weight vector q is not estimated. If dof = NULL or a numeric vector, then dof is selected by K-fold CV automatically and q is estimated.
- `lambda` If NULL or a vector of length >=2, it is selected by KCV.
- `lambda_var` If NULL or a vector of length >=2, it is selected by KCV.
- `prior_type` "NCJ" for non-conjugate prior and "CJ" for conjugate prior for scale matrix Sigma.
- `num_folds` Number of folds for KCV.
- `m0` A hyperparameter for inverse Wishart distribution for Sigma

Details

The shrinkage parameters, lambda and lambda_var, for the semiparametric Bayesian shrinkage estimator are selected by KCV. See help(lm_semi_Bayes_PCV) for details about semiparametric Bayesian estimator.

References

Description

Returns the log-likelihood of a VAR model estimated by VARshrink. It extends vars::logLik.varest() to incorporate 1) multivariate t-distribution for residuals, 2) scale matrix Sigma provided by shrinkage methods, and 3) effective number of parameters provided by shrinkage methods.

Usage

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

Arguments

object An object of class "varshrinkest"
...
Currently not used.

Details

Acknowledgement: This code was contributed by Sung-Hoon Han & Dong-Han Lee @ Kangwon National University (2018.11.29.)

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
logLik(estim)

normality.test_sh Normality, multivariate skewness and kurtosis test

Description

This function computes univariate and multivariate Jarque-Bera tests and multivariate skewness and kurtosis tests for the residuals of a VAR(p) or of a VECM in levels. This is a modification of vars::normality.test() for the class "varshrinkest".

Usage

normality.test_sh(x, multivariate.only = TRUE)
Arguments

x An object of class "varshrinkest" obtained by VARshrink().
multivariate.only
    If TRUE, only the multivariate statistics is computed.

See Also

normality.test

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
normality.test_sh(estim)

Phi.varshrinkest Coefficient matrices of the MA representation

Description

Returns the estimated coefficient matrices of the moving average representation of a stable VAR(p),
of an SVAR as an array or a converted VECM to VAR. This is a modification of vars::Phi() for the
class "varshrinkest".

Usage

## S3 method for class 'varshrinkest'
Phi(x, nstep = 10, ...)

Arguments

x An object of class 'varshrinkest', generated by VARshrink().
nstep An integer specifying the number of moving error coefficient matrices to be calculated.
    ... Currently not used.

See Also

Phi
Description

Forecasting a VAR object of class 'varshrinkest' with confidence bands. This is a modification of \texttt{vars::predict.varest()} for the class "varshrinkest".

Usage

\begin{verbatim}
## S3 method for class 'varshrinkest'
predict(object, ..., n.ahead = 10, ci = 0.95, 
dumvar = NULL)
\end{verbatim}

Arguments

\begin{itemize}
\item \textbf{object} \hspace{1cm} An object of class 'varshrinkest'; generated by \texttt{VARshrink()}
\item \textbf{...} \hspace{1cm} currently not used.
\item \textbf{n.ahead} \hspace{1cm} An integer specifying the number of forecast steps.
\item \textbf{ci} \hspace{1cm} The forecast confidence interval
\item \textbf{dumvar} \hspace{1cm} Matrix for objects of class 'vec2var' or 'varest', if the \texttt{dumvar} argument in \texttt{ca.jo()} has been used or if the \texttt{exogen} argument in \texttt{VARshrink()} has been used, respectively. The matrix should have the same column dimension as in the call to \texttt{ca.jo()} or to \texttt{VARshrink()} and row dimension equal to \texttt{n.ahead}.
\end{itemize}

Description

Print method for an object of class "varshrinkest"

Usage

\begin{verbatim}
## S3 method for class 'varshrinkest'
print(x, digits = max(3, getOption("digits") - 3), 
...)
\end{verbatim}

Arguments

\begin{itemize}
\item \textbf{x} \hspace{1cm} An object of class "varshrinkest"
\item \textbf{digits, ...} \hspace{1cm} Other arguments for \texttt{print()} method
\end{itemize}
Examples

```r
data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
print(estim)
```

print.varshsum  

*Print method for class "varshsum"*

Description

Print method for an object obtained by summary.varshrinkest().

Usage

```r
## S3 method for class 'varshsum'
print(x, digits = max(3, getOption("digits") - 3),
     signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

- `x`: An object of class "varshsum"
- `digits`, `signif.stars`, `...`: Other arguments for print(), printCoefmat(), format() method

Details

This function extends print.varsum() for VAR models estimated by shrinkage methods. The output includes scale matrix Sigma and degree of freedom dof for multivariate t-distribution for residuals.

restrict_sh  

*Restricted VAR*

Description

This is a modification of vars::restrict() for the class "varshrinkest". Warning: THIS CODE IS NOT COMPLETE: this function may raise an error because it ignores shrinkage estimation.

Usage

```r
restrict_sh(x, ...)
```

Arguments

- `x`: An object of class "varshrinkest"
- `...`: Other arguments to vars::restrict()
roots_sh

See Also

restrict

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
restrict_sh(estim)

roots_sh

Eigenvalues of the companion coefficient matrix of a VAR(p)-process

Description

This is a variant of vars::roots() for an object of class `varshrinkest`, VAR parameters estimated by VARshrink().

Usage

roots_sh(x, modulus = TRUE)

Arguments

x
An object of class "varshrinkest"

modulus
TRUE for modulus of the roots.

See Also

roots

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
roots_sh(estim)
serial.test_sh  
*Test for serially correlated errors for VAR shrinkage estimate*

**Description**

An extension of `vars::serial.test()` to the class "varshrinkest".

**Usage**

```r
serial.test_sh(x, lags.pt = 16, lags.bg = 5,
               type = c("PT.asymptotic", "PT.adjusted", "BG", "ES"))
```

**Arguments**

- `x`  
  An object of class "varshrinkest" obtained by `VARshrink()`.

- `lags.pt`, `lags.bg`, `type`  
  Other arguments for `vars::serial.test()`. see help(serial.test) for details.

**See Also**

- `serial.test`

**Examples**

```r
data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
serial.test_sh(estim)
```

shrinkVARcoef  
*Semiparametric Bayesian Shrinkage Estimator for Multivariate Regression*

**Description**

Compute the semiparametric Bayesian shrinkage estimator of Psi and Sigma for a given shrinkage parameter lambda. The function is a private function for `lm_semi_Bayes_PCV()` and `lm_ShVAR_KCV()`.

**Usage**

```r
shrinkVARcoef(Y, X, lambda, dof = Inf, prior_type = "NCJ",
               TolDRes = 1e-04, m0 = ncol(Y))
```
Arguments

- **Y**: An N x K matrix of dependent variables.
- **X**: An N x M matrix of regressors.
- **lambda**: A shrinkage intensity parameter value between 0~1.
- **dof**: Degree of freedom for multivariate t-distribution. If NULL or Inf, then use multivariate normal distribution.
- **prior_type**: "NCJ" for non-conjugate prior and "CJ" for conjugate prior for scale matrix Sigma.
- **TolDRes**: Tolerance parameter for stopping criterion.
- **m0**: A hyperparameter for inverse Wishart distribution for Sigma.

References


---

**simVARmodel**

*Generate multivariate time series data using the given VAR model*

**Description**

Generate a multivariate time series data set using the given VAR model.

**Usage**

```r
simVARmodel(numT, model, burnin = 0)
```

**Arguments**

- **numT**: Number of observed time points, T.
- **model**: A list object with Coef, Sigma, dof; Coef is a list with A and c; A is a list object of K-by-K coefficient matrices and c is a length-K vector. Sigma is a K-by-K scale matrix and dof is a degree of freedom for multivariate t-distribution for noise.
- **burnin**: Number of initial points which are not included in the final values.

**Details**

First, it creates (p+burnin+numT x K) data, then it remove the first (p+burnin) vectors. Finally, it returns (numT x K) data.

**Value**

A numT-by-K matrix
Examples

```r
myCoef <- list(A = list(matrix(c(0.5, 0, 0, 0.5), 2, 2)), c = c(0.2, 0.7))
myModel <- list(Coef = myCoef, Sigma = diag(0.1^2, 2), dof = Inf)
simVARmodel(numT = 100, model = myModel, burnin = 10)
```

stableity_sh

Stability function

Description

A variant of vars::stability(). Warning: this function has not been tested for small sample sizes yet.

Usage

```r
stableity_sh(x, type = c("OLS-CUSUM", "Rec-CUSUM", "Rec-MOSUM", "OLS-MOSUM", "RE", "ME", "Score-CUSUM", "Score-MOSUM", "fluctuation"), h = 0.15, dynamic = FALSE, rescale = TRUE, ...)
```

Arguments

- `x`: An object of class "varshrinkest"
- `type`, `h`, `dynamic`, `rescale`, ...
- Other arguments to strucchange::efp()

See Also

`stability`

Examples

```r
data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
stableity_sh(estim)
```

Summary.shrinklm

Summary method for class "shrinklm"

Description

Class "shrinklm" inherits the class "lm", and it extends the "lm" class to incorporate shrinkage estimates with effective number of parameter.
summary.varshrinkest

Usage

## S3 method for class 'shrinklm'
summary(object, correlation = FALSE,
        symbolic.cor = FALSE, ...)

Arguments

object          An object of class "shrinklm"
correlation     If TRUE, the correlation matrix of the estimated coefficients is returned and printed.
symbolic.cor   If TRUE, print the correlations in a symbolic form rather than as numbers
...
     Currently not used.

summary.varshrinkest  Summary method for an object of class 'varshrinkest', VAR parameters estimated by VARshrink()

Description

Extend summary.varest() to class 'varshrinkest' to incorporate adapted methods for new classes: summary.shrinklm(), logLik.varshrinkest(), roots.varshrinkest().

Usage

## S3 method for class 'varshrinkest'
summary(object, equations = NULL, ...)

Arguments

object          An object of class "varshrinkest", usually a result of call to "VARshrink()".
equations       Subset of names of endogenous time series variables to summarize.
...
     Currently not used.

Details

Code is modified to avoid call to data matrices ($y, $datamat) and to use effective numbers of parameters of shrinkage estimates.

Output includes the scale matrix, Sigma, and degree-of-freedom, dof, for multivariate t-distribution for residuals.

Examples

data(Canada, package = "vars")
y <- diff(Canada)
estim <- VARshrink(y, p = 2, type = "const", method = "ridge")
summary(estim)
**VARshrink**  

*Shrinkage estimation of VAR parameters*

---

**Description**

Shrinkage estimation methods for high-dimensional VAR models. Consider VAR(p) model: \( y_t = A_1 y_{t-1} + \ldots + A_p y_{t-p} + C d_t + e_t \), where \( y_t \) is K-dimensional time series, \( d_t \) is deterministic regressors, \( e_t \) is a noise process, and \( A_1, \ldots, A_p, \) and \( C \) are coefficient matrices. Exogenous variables can be included additionally as regressors.

**Usage**

```r
VARshrink(y, p = 1, type = c("const", "trend", "both", "none"),
season = NULL, exogen = NULL, method = c("ridge", "ns", "fbayes",
"sbayes", "kcv"), lambda = NULL, lambda_var = NULL, dof = Inf, ...)
```

**Arguments**

- **y**: A T-by-K matrix of endogenous variables  
- **p**: Integer for the lag order  
- **type**: Type of deterministic regressors to include. # 1) "const" - the constant. 2) "trend" - the trend. 3) "both" - both the constant and the trend. 4) "none" - no deterministic regressors. ***Note: In the package version <= 0.3, method='ns' does not accept type="const" and type="both" to avoid constant term.***  
- **season**: An integer value of frequency for inclusion of centered seasonal dummy variables. abs(season) >= 3.  
- **exogen**: A T-by-L matrix of exogenous variables. Default is NULL.  
- **method**: 1) "ridge" - multivariate ridge regression. 2) "ns" - a Stein-type nonparametric shrinkage method. 3) "fbayes" - a full Bayesian shrinkage method using noninformative priors. 4) "sbayes" - a semiparametric Bayesian shrinkage method using parameterized cross validation. 5) "kcv" - a semiparametric Bayesian shrinkage method using K-fold cross validation  
- **lambda, lambda_var**: Shrinkage parameter value(s). Use of this parameter is slightly different for each method: the same value does not imply the same shrinkage estimates.  
- **dof**: Degree of freedom of multivariate t-distribution for noise. Valid only for method = "fbayes" and method = "sbayes". dof=Inf means multivariate normal distribution.  
- **...**: Extra arguments to pass to a specific function of the estimation method. For example, burnincycle and mcmccycle are for "fbayes".

**Details**

Shrinkage estimation methods can estimate the coefficients even when the dimensionality K is larger than the number of observations.
VARshrink

Value

An object of class "varshrinkest" with the components: varresult, datamat, y, type, p, K, obs, totobs, restrictions, method, lambda, call. The class "varshrinkest" inherits the class "varest" in the package vars.

Examples

data(Canada, package = "vars")
y <- diff(Canada)
VARshrink(y, p = 2, type = "const", method = "ridge")
Index

Acoef, 3
Acoef_sh, 2
arch.test, 3
arch.test_sh, 3

Bcoef, 4
Bcoef_sh, 4
BQ, 5
BQ_sh, 5
calcSSE_Acoef, 5
causality, 6
causality_sh, 6
convPsi2varresult, 7
createVARCoefs_ltriangular, 8

fevd, 9
fevd.varshrinkest, 9

irf, 10
irf.varshrinkest, 10

lm_full_Bayes_SR, 11
lm_multiv_ridge, 12
lm_semi_Bayes_PCV, 13
lm_ShVAR_KCV, 14
logLik.varshrinkest, 15

normality.test, 16
normality.test_sh, 15

Phi, 16
Phi.varshrinkest, 16
predict.varshrinkest, 17
print.varshrinkest, 17
print.varshsum, 18

restrict, 19
restrict_sh, 18
roots, 19
roots_sh, 19

serial.test, 20
serial.test_sh, 20
shrinkVARcoef, 20
simVARmodel, 21
stability, 22
stability_sh, 22
summary.shrinklm, 22
summary.varshrinkest, 23

VARshrink, 24

26