Package ‘BigVAR’

January 9, 2023

Type Package
Title Dimension Reduction Methods for Multivariate Time Series
Version 1.1.2
Maintainer Will Nicholson <wbn8@cornell.edu>
Date 2023-01-09
Depends R (>= 3.5.0), methods, lattice
Imports MASS, zoo, Rcpp, stats, utils, grDevices, graphics, abind
License GPL (>= 2)
LazyLoad yes
SystemRequirements C++11
LinkingTo Rcpp, RcppArmadillo, RcppEigen
URL https://github.com/wbnicholson/BigVAR
BugReports https://github.com/wbnicholson/BigVAR/issues
RoxygenNote 7.2.3
Encoding UTF-8
Suggests knitr, rmarkdown, gridExtra, expm, MCS, quantmod, codetools
VignetteBuilder knitr
NeedsCompilation yes
Author Will Nicholson [cre, aut],
David Matteson [aut],
Jacob Bien [aut],
Ines Wilms [aut]
Repository CRAN
Date/Publication 2023-01-09 14:43:19 UTC
Generator for Simulated Multivariate Time Series

Description

Coefficient matrix for a stationary simulated multivariate time series

Details

Example generator matrix adapted from Table 3.2 of Gredenhoff and Karlsson (1997)

Author(s)

Will Nicholson

References

Description

BigVAR implements the HLAG and VARX-L frameworks which allow for the estimation of vector autoregressions and vector autoregressions with exogenous variables using structured convex penalties. This package originated as a 2014 Google "Summer of Code" Project. The development version of this package is hosted on github: https://github.com/wbnicholson/BigVAR.

Details

To use the facilities of this package, starting with an $T \times k + m$ multivariate time series (in which $T$ denotes the length of the series, $k$ the number of endogenous or "model") and run constructModel to create an object of class BigVAR. cv.BigVAR creates an object of class BigVAR.results, which chooses an optimal penalty parameter based on minimizing h-step ahead forecasts on a specified cross-validation period over a grid of values as well as comparisons against AIC, BIC, unconditional mean, and a random walk. There are plot functions for both BigVAR (plot.BigVAR) and BigVAR.results (plot) as well as a predict function for BigVAR.results (predict).

Author(s)

Will Nicholson <wbn8@cornell.edu>,

References


See Also

constructModel, cv.BigVAR, BigVAR.results, plot, predict

Examples

# Fit a Basic VAR-L(3,4) on simulated data
data(Y)
T1=floor(nrow(Y)/3)
T2=floor(2*nrow(Y)/3)
m1=constructModel(Y,p=4,struct="Basic",gran=c(50,10),verbose=FALSE,T1=T1,T2=T2,IC=FALSE)
plot(m1)
results=cv.BigVAR(m1)
plot(results)
predict(results,n.ahead=1)

BigVAR-class

Description
An object class to be used with cv.BigVAR

Details
To construct an object of class BigVAR, use the function constructModel

Slots
Data a T x k multivariate time series
model_data processed time series and lag matrix
lagmax Maximal lag order for modeled series
intercept Indicator as to whether an intercept should be included
Structure Penalty Structure
Relaxed Indicator for relaxed VAR
Granularity  Granularity of penalty grid
horizon Desired Forecast Horizon
crossval Cross-Validation Procedure
Minnesota Minnesota Prior Indicator
verbose Indicator for Verbose output
dates dates extracted from an xts object
ic Indicator for including AIC and BIC benchmarks
VARX VARX Model Specifications
VARXI VARX Indicator
T1 Index of time series in which to start cross validation
T2 Index of times series in which to start forecast evaluation
ONESE Indicator for 'One Standard Error Heuristic'
ownlambdas Indicator for user-supplied lambdas
tf Indicator for transfer function
alpha Grid of candidate alpha values (applies only to Sparse VARX-L and Elastic Net models)
recursive  Indicator as to whether recursive multi-step forecasts are used (applies only to multiple
horizon VAR models)
constvec  vector indicating variables to shrink toward a random walk instead of toward zero (valid
only if Minnesota is TRUE)
tol     optimization tolerance
window.size  size of rolling window. If set to NULL an expanding window will be used.
separate_lambdas  indicator to use separate penalty parameter for each time series (default FALSE)
loss     Loss function to select penalty parameter (one of 'L1','L2','Huber').
delta     delta parameter for Huber loss (default 2.5)
gamma     gamma parameter for SCAD or MCP penalty (default 3)
rolling_oos True or False: indicator to update the penalty parameter over the evaluation period
            (default False)
linear     indicator for linearly decrementing penalty grid (FALSE is log-linear).
refit_fraction  fraction of least squares refit to incorporate (default is 1).

See Also

constructModel

---

**Description**

Fit a BigVAR object with a structured penalty (VARX-L or HLAG).

**Usage**

BigVAR.est(object)

**Arguments**

object     BigVAR object created from ConstructModel

**Details**

Fits HLAG or VARX-L model on a BigVAR object. Does not perform cross-validation. This
method allows the user to construct their own penalty parameter selection procedure.

**Value**

An array of $k \times kp \times n$ or $k \times kp + ms \times n$ coefficient matrices; one for each of the n values of
lambda.
BigVAR.fit

Simple function to fit BigVAR model with fixed penalty parameter

Description

Simple function to fit BigVAR model with fixed penalty parameter.

Usage

BigVAR.fit(
  Y,
  p,
  struct,
  lambda,
  alpha = NULL,
  VARX = list(),
  separate_lambdas = F,
  MN = F,
  C = as.double(NULL),
  intercept = TRUE,
  tf = F,
  tol = 1e-04,
  RVAR = F,
  refit_fraction = 1,
  beta = NULL,
  gamma = 3
)

Arguments

Y  $T \times k$ multivariate time series or $Y \ T \times (k + m)$ endogenous and exogenous series, respectively.

p  Predetermined maximal lag order (for modeled series).

struct  The choice of penalty structure (see details).

lambda  vector or matrix of penalty parameters.
alpha  grid of candidate parameters for the alpha in the Sparse Lag and Sparse Own/Other VAR-L
VARX  List containing VARX model specifications.
separate_lambdas  indicator for separate penalty parameters for each time series (default FALSE)
MN  Minnesota Prior Indicator
C  vector of coefficients to shrink toward a random walk (if MN is TRUE)
intercept  True or False: option to fit an intercept
tf  transfer function indicator (i.e. VARX in which p=0 & s>0) (default false)
tol  optimization tolerance (default 1e-4)
RVAR  True or False: option to refit based upon the support selected using the Relaxed-VAR procedure
refit_fraction  fraction of least squares refit to incorporate (default 1)
beta  optional \(k \times (k \times p + m \times s + 1)\) coefficient matrix to use as a 'warm start' (default NULL)
gamma  additional parameter for SCAD/MCP penalty (default 3)
@details The choices for 'struct' are as follows
• 'Basic' (Basic VARX-L)
• 'BasicEN' (Basic Elastic Net VARX-L)
• 'Lag' (Lag Group VARX-L)
• 'SparseLag' (Lag Sparse Group VARX-L)
• 'OwnOther' (Own/Other Group VARX-L)
• 'SparseOO' (Own/Other Sparse Group VARX-L)
• 'EFX' (Endogenous First VARX-L)
• 'HLAGC' (Componentwise HLAG)
• 'HLAGOO' (Own/Other HLAG)
• 'HLAGELEM' (Elementwise HLAG)
• 'Tapered' (Lag weighted Lasso VAR)
• 'BGR' (Bayesian Ridge Regression (cf. Banbura et al))
• 'MCP' (Minimax Concave Penalty (cf. Brebey and Huang))
• 'SCAD' (Smoothly Clipped Absolute Deviation (cf. Brebey and Huang))

VARX specifications consist of a list with entry k denoting the series that are to be modeled and entry s to denote the maximal lag order for exogenous series. The argument alpha is ignored unless the structure choice is 'SparseLag' or 'Lag.' By default 'alpha' is set to NULL and will be initialized as \(1/(k+1)\) in cv.BigVAR and BigVAR.est. Any user supplied values must be between 0 and 1.

Note
The specifications 'Basic,' 'Lag,' 'SparseLag,' 'SparseOO,' and 'OwnOther' can accommodate both VAR and VARX models. EFX only applies to VARX models. 'HLAGC,' 'HLAGOO,' 'HLAGELEM,' and 'Tapered' can only be used with VAR models. Our implementation of the SCAD and MCP penalties is heavily influenced by the implementation in ncvreg.
References


See Also

cv.BigVAR,BigVAR.est,constructModel

Examples

# VARX Example
# Fit a Basic VARX-L with k=2, m=1, s=2, p=4, lambda=1e-2
VARX=list()
VARX$k=2 # indicates that the first two series are modeled
VARX$s=2 # sets 2 as the maximal lag order for exogenous series
data(Y)
BigVAR.fit(Y,p=4,'Basic',lambda=1e-2,VARX=VARX)

BigVAR.intermediate This class contains the in-sample results for cv.BigVAR

Description

It inherits the class BigVAR, and contains the results from rolling validation

Fields

ZFull List containing full lag matrix and time series
InSampMSFE In-sample MSFE from optimal value of lambda
LambdaGrid Grid of candidate lambda values
index Index order of optimal lambda value
OptimalLambda Value of lambda that minimizes MSFE
Data a $T \times k$ or $T \times k + m$ multivariate time Series
lagmax Maximal lag order
Structure Penalty structure
Relaxed Indicator for relaxed VAR
Granularity  Granularity of penalty grid
horizon  Desired forecast horizon
crossval  Cross-Validation procedure
alpha  additional penalty parameter for Sparse Lag Group or Sparse Own/Other methods. Will contain either the heuristic choice of $1/(k+1)$ or the value selected by cross validation if the argument dual is set to TRUE
Minnesota  Minnesota Prior Indicator
verbose  verbose indicator
dual  indicator as to whether dual cross validation was conducted
contemp  indicator if contemporaneous exogenous predictors are used

Note
One can also access any object of class BigVAR from BigVAR.intermediate

Author(s)
Will Nicholson

BigVAR.results  This class contains the results from cv.BigVAR.

Description
It inherits the class BigVAR, but contains substantially more information.

Fields

InSampMSFE  In-sample MSFE from optimal value of lambda
LambdaGrid  Grid of candidate lambda values
index  Rank of optimal lambda value
OptimalLambda  Value of lambda that minimizes MSFE
OOSMSFE  Average Out of sample MSFE of BigVAR model with optimal lambda
seoosfmsfe  Standard error of out of sample MSFE of BigVAR model with optimal lambda
MeanMSFE  Average out of sample MSFE of unconditional mean forecast
MeanSD  Standard error of out of sample MSFE of unconditional mean forecast
MeanPreds  predictions from conditional mean model
RWMSFE  Average out of sample MSFE of random walk forecast
RWPreds  Predictions from random walk model
RWSD  Standard error of out of sample MSFE of random walk forecast
AICMSFE  Average out of sample MSFE of AIC forecast
BigVAR.results

AICSD  Standard error of out of sample MSFE of AIC forecast
AICPreds  Predictions from AIC VAR/VARX model
AICpvec  Lag orders selected from AIC VAR model
AICpvec  Lag orders selected from AIC VARX model
BICMSFE  Average out of sample MSFE of BIC forecast
BICSD  Standard error of out of sample MSFE of BIC forecast
BICPreds  Predictions from BIC VAR/VARX model
BICpvec  Lag orders selected from BIC VAR model
BICpvec  Lag orders selected from BIC VARX model
betaPred  The final estimated \( k \times kp + ms + 1 \) coefficient matrix, to be used for prediction
Zvals  The final lagged values of \( Y \), to be used for prediction
fitted  fitted values obtained from betaPred
resids  residuals obtained from betaPred
Data  a \( T \times k \) or \( T \times k + m \) multivariate time Series
lagmax  Maximal lag order
Structure  Penalty structure
Relaxed  Indicator for relaxed VAR
Granularity  Granularity of penalty grid
horizon  Desired forecast horizon
crossval  Cross-Validation procedure
alpha  additional penalty parameter for Sparse Lag Group or Sparse Own/Other methods. Will contain either the heuristic choice of \( 1/(k + 1) \) or the value selected by cross validation if the argument dual is set to TRUE
VARXI  VARX Indicator
Minnesota  Minnesota Prior Indicator
verbose  verbose indicator
dual  indicator as to whether dual cross validation was conducted
contemp  indicator if contemporaneous exogenous predictors are used
lagmatrix  matrix of lagged values used to compute residuals (of which Zvals is the final column)
betaArray  array of VAR/VARX coefficients from out of sample forecasts
sparse_count  average fraction of active coefficients in validation period
lambda_evolve_path  evolution of lambda over evaluation period

Note

One can also access any object of class BigVAR from BigVAR.results

Author(s)

Will Nicholson
**coef**

Default coef method BigVAR-results, returns the last coefficient matrix from the evaluation period

**Description**

Default coef method BigVAR-results, returns the last coefficient matrix from the evaluation period

**Usage**

```r
## S4 method for signature 'BigVAR.results'
coef(object)
```

**Arguments**

**object**  
BigVAR.results object created from cv.BigVAR

**Details**

displays formatted coefficient matrix

---

**constructModel**

Construct an object of class BigVAR

**Description**

Construct an object of class BigVAR

**Usage**

```r
constructModel(
  Y,  
p,  
struct,  
gran,
  h = 1,
  cv = "Rolling",
  verbose = TRUE,
  IC = TRUE,
  VARX = list(),
  T1 = floor(nrow(Y)/3),
  T2 = floor(2 * nrow(Y)/3),
  ONESE = FALSE,
  ownlambdas = FALSE,
  recursive = FALSE,
  dates = as.character(NULL),
```
window.size = 0,
separate_lambdas = FALSE,
linear = FALSE,
loss = "L2",
rolling_oos = FALSE,
model.controls = list()
)

Arguments

Y \quad T \times k \text{ multivariate time series or } Y T \times (k + m) \text{ endogenous and exogenous series, respectively.}
p \quad \text{Predetermined maximal lag order (for modeled series).}
struct \quad \text{The choice of penalty structure (see details).}
gran \quad \text{vector of penalty parameter specifications.}
h \quad \text{Desired forecast horizon.}
cv \quad \text{Cross-validation approach, either 'Rolling' for rolling cross-validation or 'LOO' for leave-one-out cross-validation. 'None' for use with BigVAR.fit.}
verbose \quad \text{Verbose output while estimating.}
IC \quad \text{True or False: whether to include AIC and BIC benchmarks.}
VARX \quad \text{List containing VARX model specifications.}
T1 \quad \text{Index of time series in which to start cross validation.}
T2 \quad \text{Index of times series in which to start forecast evaluation.}
ONESE \quad \text{True or False: whether to use the 'One Standard Error Heuristic.'}
ownlambdas \quad \text{True or False: Indicator for user-supplied penalty parameters.}
recursive \quad \text{True or False: Indicator as to whether iterative multi-step predictions are desired in the VAR context if the forecast horizon is greater than 1.}
dates \quad \text{optional vector of dates corresponding to } Y .
window.size \quad \text{size of rolling window. If set to 0 an expanding window will be used.}
separate_lambdas \quad \text{indicator for separate penalty parameters for each time series (default FALSE).}
linear \quad \text{indicator for linearly decrementing penalty grid (FALSE is log-linear; default FALSE).}
loss \quad \text{Loss function to select penalty parameter (one of 'L1','L2','Huber')}
rolling_oos \quad \text{True or False: indicator to update the penalty parameter over the evaluation period (default False)}
model.controls \quad \text{named list of control parameters for BigVAR model estimation (see details).}

Details

The choices for 'struct' are as follows

- 'Basic' (Basic VARX-L)
• 'BasicEN' (Elastic Net VARX-L)
• 'Lag' (Lag Group VARX-L)
• 'SparseLag' (Lag Sparse Group VARX-L)
• 'OwnOther' (Own/Other Group VARX-L)
• 'SparseOO' (Own/Other Sparse Group VARX-L)
• 'EFX' (Endogenous First VARX-L)
• 'HLAGC' (Componentwise HLAG)
• 'HLAGOO' (Own/Other HLAG)
• 'HLAGELEM' (Elementwise HLAG)
• 'Tapered' (Lag weighted Lasso VAR)
• 'BGR' (Bayesian Ridge Regression (cf. Banbura et al))
• 'MCP' (Minimax Concave Penalty (cf. Breheny and Huang))
• 'SCAD' (Smoothly Clipped Absolute Deviation Penalty (cf. Breheny and Huang))

The first number in the vector ‘gran’ specifies how deep to construct the penalty grid and the second specifies how many penalty parameters to use. If ownlambas is set to TRUE, gran should contain the user-supplied penalty parameters.

VARX specifications consist of a named list with entry k denoting the series that are to be modeled and entry s to denote the maximal lag order for exogenous series.

As the capabilities of BigVAR have expanded, we have decided to consolidate parameters in the list model.controls. These parameters include:

• 'alpha:' grid of candidate parameters for the alpha in the Basic Elastic Net, Sparse Lag, Sparse Own/Other VARX-L.
• 'C:' vector of coefficients to shrink toward a random walk (if MN is TRUE).
• 'delta:' parameter for Huber loss (default 2.5)
• 'intercept:' option to fit an intercept, default TRUE
• 'loss:' Loss function to select penalty parameter (one of 'L1', 'L2', 'Huber')
• 'MN:' Minnesota Prior Indicator, default FALSE
• 'RVAR:' option to refit based upon the support selected using the Relaxed-VAR procedure (default FALSE).
• 'refit_fraction:' If RVAR is TRUE, proportional tradeoff between least squares fit and penalized fit (default 1).
• 'tol:' optimization tolerance (default 1e-4)

The argument alpha is ignored unless the structure choice is 'SparseLag' or 'Lag.' By default 'alpha' is set to NULL and will be initialized as 1/(k+1) in cv.BigVAR and BigVAR.est. Any user supplied values must be between 0 and 1.

Note

The specifications 'Basic', 'BasicEN', 'Lag', 'SparseLag', 'SparseOO', 'OwnOther', 'MCP', and 'SCAD' can accommodate both VAR and VARX models. EFX only applies to VARX models. 'HLAGC', 'HLAGOO', 'HLAGELEM,' and 'Tapered' can only be used with VAR models. Our implementation of the SCAD and MCP penalties is heavily influenced by the package ncvreg.
References


See Also

cv.BigVAR,BigVAR.est

Examples

# VARX Example
# Create a Basic VARX-L with k=2, m=1, s=2, p=4
VARX=list()
VARX$k=2 # indicates that the first two series are modeled
VARX$s=2 # sets 2 as the maximal lag order for exogenous series
data(Y)
T1=floor(nrow(Y)/3)
T2=floor(2*nrow(Y)/3)
Model1=constructModel(Y,p=4,struct='Basic',gran=c(50,10),verbose=FALSE,VARX=VARX,T1=T1,T2=T2)

cv.BigVAR

Cross Validation for BigVAR

Description

Cross Validation for BigVAR

Usage

cv.BigVAR(object)

Arguments

object BigVAR object created from ConstructModel

Details

The main function of the BigVAR package. Performs cross validation to select penalty parameters over a training sample (as the minimizer of in-sample MSFE), then evaluates them over a test set. Compares against sample mean, random walk, AIC, and BIC benchmarks. Creates an object of class BigVAR.results
MultVarSim

Value
An object of class BigVAR.results.

See Also
constructModel, BigVAR.results, BigVAR.est

Examples

data(Y)
# Fit a Basic VAR-X-L with rolling cross validation
Model1=constructModel(Y,p=4,struct='Basic',gran=c(50,10), verbose=FALSE)
results=cv.BigVAR(Model1)

MultVarSim

Simulate a VAR

Description
Simulate a VAR

Usage
MultVarSim(k, A1, p, Sigma, T)

Arguments

- **k**: Number of Series
- **A1**: Either a $k \times k$ coefficient matrix or a $kp \times kp$ matrix created using VarptoVar1MC.
- **p**: Maximum Lag Order
- **Sigma**: Residual Covariance Matrix of dimension $k \times k$
- **T**: Number of simulations

Value
Returns a $T \times k$ of realizations from a VAR.

References
Lutkepohl, 'A New Introduction to Multiple Time Series Analysis'

See Also
VarptoVar1MC
Examples

```r
k=3;p=6
B=matrix(0,nrow=k,ncol=p*k)
A1<- matrix(c(.4,-.02,.01,-.02,.3,.02,.01,.04,.3),ncol=3,nrow=3)
A2 <- matrix(c(.2,0,0,.3,0,0,.13),ncol=3,nrow=3)
B[,1:k]=A1
B[(4*k+1):((5*k)]]=A2
A <- VarptoVar1MC(B,p,k)
Y <- MultVarSim(k,A,p,.1*diag(k),100)
```

```r
plot
Plot an object of class BigVAR.results
```

Description

Plot an object of class BigVAR.results

Usage

```r
## S4 method for signature 'BigVAR.results'
plot(x, y = NULL, ...)
```

Arguments

- **x**: BigVAR.results object created from cv.BigVAR
- **y**: NULL
- **...**: additional arguments

Details

Plots the in sample MSFE of all values of lambda with the optimal value highlighted.

```r
plot.BigVAR
Plot a BigVAR object
```

Description

Plot a BigVAR object

Usage

```r
## S4 method for signature 'BigVAR'
plot(x, y = NULL, ...)
```
Arguments

- **x**: BigVAR object created from `ConstructModel`
- **y**: NULL
- **...**: additional plot arguments

Details

Uses `plot.zoo` to plot each individual series of `Y` on a single plot.

Value

NA, side effect is graph

See Also

- `constructModel`

---

**predict**  
*Forecast using a BigVAR.results object*

Description

Forecast using a BigVAR.results object

Usage

```r
predict(object,...)
```

Arguments

- **object**: BigVAR.results object from `cv.BigVAR`
- **...**: additional arguments affecting the predictions produced (e.g. `n.ahead`, `confint`)

Details

Provides `n.ahead` step forecasts using the model produced by `cv.BigVAR`. If `confint` is set to `TRUE`, a 95 percent confidence interval will also be returned.

See Also

- `cv.BigVAR`

Examples

```r
data(Y)
Y=Y[1:100,]
Model1=constructModel(Y,p=4,struct='Basic',gran=c(50,10),verbose=FALSE)
results=cv.BigVAR(Model1)
predict(results,n.ahead=1)
```
PredictVARX  
*One-step ahead predictions for VARX models*

**Description**

One-step ahead predictions for VARX models

**Usage**

```r
PredictVARX(VARXRes)
```

**Arguments**

- `VARXRes` the results from `VARXFit`

**Value**

Returns a vector consisting of the out-of-sample forecasts for the provided `VARXFit` model.

**See Also**

`VARXFit`

**Examples**

```r
data(Y)
# fit a VAR_3(3)
mod <- VARXFit(Y,3,NULL,NULL)
pred <- PredictVARX(mod)
```

---

**show**  
*Default show method for an object of class BigVAR.results*

**Description**

Default show method for an object of class BigVAR.results

**Usage**

```r
## S4 method for signature 'BigVAR.results'
show(object)
```

**Arguments**

- `object` BigVAR.results object created from `cv.BigVAR`
**Details**

prints forecast results and additional diagnostic information as well as comparisons with mean, random walk, and AIC, and BIC benchmarks

**See Also**

cv.BigVAR, BigVAR.results

---

**show.BigVAR**

Default show method for an object of class BigVAR

**Description**

Default show method for an object of class BigVAR

**Usage**

```r
## S4 method for signature 'BigVAR'
show(object)
```

**Arguments**

- `object` BigVAR object created from ConstructModel

**Value**

Displays the following information about the BigVAR object:

- Prints the first 5 rows of Y
- Penalty Structure
- Relaxed Least Squares Indicator
- Maximum lag order
- VARX Specifications (if applicable)
- Start, end of cross validation period

**See Also**

constructModel
SparsityPlot.BigVAR.results

Sparsity Plot of a BigVAR.results object

Description

Sparsity Plot of a BigVAR.results object

Usage

SparsityPlot.BigVAR.results(object)

Arguments

object BigVAR.results object

Details

Uses levelplot from the lattice package to plot the magnitude of each coefficient in the last coefficient estimated by cv.BigVAR.

Value

NA, side effect is graph

See Also

cv.BigVAR, BigVAR.results

Examples

data(Y)
Y <- Y[1:100,]
Model1 <- constructModel(Y,p=4,struct='Basic',gran=c(50,10),verbose=FALSE)
SparsityPlot.BigVAR.results(cv.BigVAR(Model1))

VarptoVar1MC

Converts a VAR coefficient matrix of order p to multiple companion form

Description

Converts a VAR coefficient matrix of order p to multiple companion form

Usage

VarptoVar1MC(B, p, k)
VARXFit

Arguments

- **B**: a $k \times kp$ coefficient matrix
- **p**: Lag order
- **k**: Number of Series

Value

Returns a $kp \times kp$ coefficient matrix representing all coefficient matrices contained in $A_i$ as a VAR(1).

References

See page 15 of Lutkepohl, 'A New Introduction to Multiple Time Series Analysis'

See Also

- MultVarSim

Examples

```r
k=3; p=6
B=matrix(0,nrow=k,ncol=p*k)
A1<- matrix(c(.4,-.02,.01,-.02,.3,.02,.01,.04,.3),ncol=3,nrow=3)
A2 <- matrix(c(.2,0,0,0,.3,0,0,0,.13),ncol=3,nrow=3)
B[,1:k]=A1
B[(4*k+1):(5*k)]=A2
A <- VarptoVar1MC(B,p,k)
```

VARXFit

**Fit a VAR or VARX model by least squares**

Description

Fit a VAR or VARX model by least squares

Usage

```r
VARXFit(Y, p, IC, VARX = NULL)
```

Arguments

- **Y**: a $t \times k$ multivariate time series
- **p**: maximum lag order
- **IC**: Information criterion indicator, if set to NULL, it will fit a least squares VAR(X) of orders p and s. Otherwise, if set to 'AIC' or 'BIC' it return the model with lag orders that minimize the given IC.
- **VARX**: a list of VARX specifications (as in `constructModel` (or NULL))
Details

This function uses a modified form of the least squares technique proposed by Neumaier and Schneider (2001). It fits a least squares VAR or VARX via a QR decomposition that does not require explicit matrix inversion. This results in improved computational performance as well as numerical stability over the conventional least squares approach.

Value

Returns a list with four entries:

- 'Bhat' Estimated $k \times kp + ms$ coefficient matrix
- 'SigmaUEstimated' $k \times k$ residual covariance matrix
- 'phat' Selected lag order for VAR component
- 'shat' Selected lag order for VARX component
- 'Y' multivariate time series retained for prediction purposes
- 'Y' number of endogenous (modeled) time series

References


See Also

constructModel, cv.BigVAR, BigVAR.fit

Examples

data(Y)
# fit a VAR_3(3)
mod <- VARXFit(Y,3,NULL,NULL)
# fit a VAR_3 with p= 6 and lag selected according to AIC
modAIC <- VARXFit(Y,6,'AIC',NULL)
# Fit a VARX_{2,1} with p=6, s=4 and lags selected by BIC
modXBIC <- VARXFit(Y,6,'BIC',list(k=1,s=4))

---

VARXForecastEval

Evaluate forecasts from a VAR or VARX with lag orders selected by AIC/BIC

Description

Evaluate forecasts from a VAR or VARX with lag orders selected by AIC/BIC
Usage

VARXForecastEval(
    Y,
    X,
    p,
    s,
    T1,
    T2,
    IC,
    h,
    iterated = FALSE,
    loss = "L2",
    delta = 2.5
)

Arguments

Y    a $T \times k$ multivariate time series
X    a $T \times m$ multivariate time series of unmodeled exogenous variables
p    maximum lag order for endogenous series
s    maximum lag order for exogenous series
T1   start of forecast evaluation period.
T2   end of forecast evaluation period
IC   specifies whether to select lag order according to 'AIC' or 'BIC'
h    desired forecast horizon
iterated indicator as to whether to use iterated or direct multistep forecasts (if applicable, VAR context only)
loss loss function (default 'L2', one of 'L1','L2','Huber')
delta delta for Huber loss function (default 2.5)

Details

This function evaluates the one-step ahead forecasts of a VAR or VARX fit by least squares over an evaluation period. At every point in time, lag orders for the endogenous and exogenous series are selected according to AIC or BIC. This function is run automatically when cv.BigVAR is called unless ic is set to FALSE in constructModel.

Value

Returns the one-step ahead MSFE as well as the forecasts over the evaluation period and lag order selected.

References

VARXLagCons

Construct a VAR or VARX lag matrix

Usage

VARXLagCons(Y, X = NULL, p, s = 0, oos = FALSE, contemp = FALSE)

Arguments

Y  a $T \times k$ matrix of endogenous (modeled) series
X  a $T \times m$ matrix of exogenous (unmodeled) series (default NULL)
p  Endogenous Lag order
s  exogenous lag order (default zero)
oos  indicator as to whether the data should be constructed for out of sample prediction (i.e. last available entries of Y as final lags default FALSE)
contemp  indicator as to whether to use contemporaneous exogenous predictors (for example, if exogenous series become available before exogenous default FALSE).

Details

This function is not required unless you which to design your own cross validation routine.

Value

list with two entries:

- 'Z' $kp + ms + 1 \times T - \max(p, s)$ VARX lag matrix
- 'Y' adjusted $k \times T - \max(p, s)$ endogenous series

See Also

VARXFit, constructModel, cv.BigVAR

Examples

data(Y)

# Evaluate the performance of a VAR with lags selected by BIC.
p <- 4
T1 <- floor(nrow(Y))/3
T2 <- floor(2*nrow(Y))/3
# Matrix of zeros for X
X <- matrix(0,nrow=nrow(Y),ncol=ncol(Y))
BICMSFE <- VARXForecastEval(Y,X,p,0,T1,T2,'BIC',1)
Y

References
See page 15 of Lutkepohl, 'A New Introduction to Multiple Time Series Analysis

See Also
MultVarSim

Examples

data(Y)
# construct VAR lag matrix with p=4
ZZ<-VARXLagCons(Y,X=NULL,p=4,s=0)

<table>
<thead>
<tr>
<th>Y</th>
<th>Simulated Multivariate Time Series</th>
</tr>
</thead>
</table>

Description
Realization of a simulated multivariate time series

Details
100 × 3 multivariate time series distributed according to the generator matrix A.

Author(s)
Will Nicholson
## Index

<table>
<thead>
<tr>
<th>Page</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 25</td>
<td>A</td>
</tr>
<tr>
<td>3, 3</td>
<td>BigVAR</td>
</tr>
<tr>
<td>4</td>
<td>BigVAR-class</td>
</tr>
<tr>
<td>5, 8, 14, 15</td>
<td>BigVAR.est</td>
</tr>
<tr>
<td>6, 22</td>
<td>BigVAR.fit</td>
</tr>
<tr>
<td>8</td>
<td>BigVAR.intermediate</td>
</tr>
<tr>
<td>8</td>
<td>BigVAR.intermediate-class</td>
</tr>
<tr>
<td>3, 6, 9, 15, 19, 20</td>
<td>BigVAR.results</td>
</tr>
<tr>
<td>9</td>
<td>BigVAR.results-class (BigVAR.results)</td>
</tr>
<tr>
<td>11</td>
<td>coef</td>
</tr>
<tr>
<td>11</td>
<td>coef, BigVAR.results-method (coef)</td>
</tr>
<tr>
<td>11</td>
<td>coef-methods (coef)</td>
</tr>
<tr>
<td>3–6, 8, 15, 17, 19, 21–24</td>
<td>constructModel</td>
</tr>
<tr>
<td>3, 6, 8, 14, 14, 17, 19, 20, 22–24</td>
<td>cv.BigVAR</td>
</tr>
<tr>
<td>14</td>
<td>cv.BigVAR, BigVAR-method (cv.BigVAR)</td>
</tr>
<tr>
<td>15, 21, 25</td>
<td>MultVarSim</td>
</tr>
<tr>
<td>3, 16</td>
<td>plot</td>
</tr>
<tr>
<td>16</td>
<td>plot, BigVAR-method (plot.BigVAR)</td>
</tr>
<tr>
<td>16</td>
<td>plot, BigVAR.results-method (plot)</td>
</tr>
<tr>
<td>16</td>
<td>plot-methods (plot)</td>
</tr>
<tr>
<td>3, 16</td>
<td>plot.BigVAR</td>
</tr>
<tr>
<td>3, 17</td>
<td>predict</td>
</tr>
<tr>
<td>17</td>
<td>predict, BigVAR.results-method</td>
</tr>
<tr>
<td>17</td>
<td>(predict)</td>
</tr>
<tr>
<td>18</td>
<td>PredictVARX</td>
</tr>
<tr>
<td>19</td>
<td>show</td>
</tr>
<tr>
<td>19</td>
<td>show, BigVAR-method (show.BigVAR)</td>
</tr>
<tr>
<td>18</td>
<td>show, BigVAR.results-method (show)</td>
</tr>
<tr>
<td>19</td>
<td>show, BigVAR</td>
</tr>
<tr>
<td>20</td>
<td>SparsityPlot.BigVAR.results</td>
</tr>
<tr>
<td>20</td>
<td>SparsityPlot.BigVAR.results, BigVAR.results-method</td>
</tr>
<tr>
<td>20</td>
<td>(SparsityPlot.BigVAR.results)</td>
</tr>
</tbody>
</table>