Package ‘BigVAR’

April 3, 2017

Type Package
Title Dimension Reduction Methods for Multivariate Time Series
Version 1.0.2
Date 2017-03-24
Description Estimates VAR and VARX models with structured Lasso Penalties.
Depends R (>= 3.1.0), methods
Imports MASS, zoo, lattice, Rcpp, stats, utils, grDevices, graphics
License GPL (>= 2)
LazyLoad yes
LinkingTo Rcpp, RcppArmadillo, RcppEigen
URL http://www.github.com/wbnicholson/BigVAR
RoxygenNote 6.0.1
NeedsCompilation yes
Author Will Nicholson [cre, aut],
  David Matteson [aut],
  Jacob Bien [aut]
Maintainer Will Nicholson <wbn8@cornell.edu>
Repository CRAN
Date/Publication 2017-04-03 06:16:57 UTC

R topics documented:

A ................................................................. 2
BigVAR .......................................................... 2
BigVAR-class .................................................. 3
BigVAR.est .................................................... 4
BigVAR.results ............................................. 5
constructModel ............................................ 6
cv.BigVAR .................................................. 8
MultVarSim .................................................. 9
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


Dimension Reduction Methods for Multivariate Time Series

Description

BigVAR implements the HVAR 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: http://www.github.com/wbnicholson/BigVAR.
BigVAR-class

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

**BigVAR Object 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 $T \times k$ multivariate time Series
- **lagmax** Maximal lag order for modeled series
- **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
  - **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 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)

See Also

- `constructModel`

---

Description

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

Usage

`BigVAR.est(object)`

Arguments

- **object** BigVAR object created from `ConstructModel`
**Details**

Fits HVAR 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.

**See Also**

constructModel, BigVAR.results, cv.BigVAR

**Examples**

data(y)
Y=Y[1:100,]
#construct a Basic VAR-L
Model1=constructModel(Y,p=4,struct="Basic",gran=c(50,10))
BigVAR.est(Model1)

---

**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
- RWSMSFE  Average out of sample MSFE of random walk forecast
- RWSD  Standard error of out of sample MSFE of random walk forecast
- AICMSFE  Average out of sample MSFE of AIC forecast
- AICSD  Standard error of out of sample MSFE of AIC forecast
- BICMSFE  Average out of sample MSFE of BIC forecast
constructModel

Construct an object of class BigVAR

Description

Construct an object of class BigVAR

Usage

```
constructModel(Y, p, struct, gran, RVAR = FALSE, h = 1, cv = "Rolling",
               MN = FALSE, verbose = TRUE, IC = TRUE, VARX = list(),
               T1 = floor(nrow(Y)/3), T2 = floor(2 * nrow(Y)/3), ONESE = FALSE,
               ownlambdas = FALSE, alpha = as.double(NULL), recursive = FALSE,
               C = as.double(NULL), dates = as.character(NULL))
```
constructModel

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).
- **gran**: vector of penalty parameter specifications.
- **RVAR**: True or False: whether to refit based upon the support selected using the Relaxed-VAR procedure.
- **h**: Desired forecast horizon.
- **cv**: Cross-validation approach, either "Rolling" for rolling cross-validation or "LOO" for leave-one-out cross-validation.
- **MN**: Minnesota Prior Indicator.
- **verbose**: Verbose output while estimating.
- **IC**: True or False: whether to include AIC and BIC benchmarks.
- **VARX**: List containing VARX model specifications.
- **T1**: Index of time series in which to start cross validation.
- **T2**: Index of times series in which to start forecast evaluation.
- **ONESE**: True or False: whether to use the "One Standard Error Heuristic".
- **ownlambdas**: True or False: Indicator for user-supplied penalty parameters.
- **alpha**: grid of candidate parameters for the alpha in the Sparse Lag and Sparse Own/Other VARX-L.
- **recursive**: 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.
- **C**: vector of coefficients to shrink toward a random walk (if MN is TRUE).
- **dates**: optional vector of dates corresponding to Y.

@details The choices for "struct" are as follows:
- "Basic" (Basic 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)
- "HVARC" (Componentwise HVAR)
- "HVAROO" (Own/Other HVAR)
- "HVARELEM" (Elementwise HVAR)
- "Tapered" (Lag weighted Lasso VAR)

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 ownlambdas is set to TRUE, gran should contain the user-supplied penalty parameters.
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. "HVARC," "HVAROO," "HVARELEM," and "Tapered" can only be used with VAR models.

References


See Also

`cv.BigVAR`, `BigVAR.est`

Examples

```r
# 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)`
MultVarSim

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

Value

An object of class BigVAR.results.

See Also

constructModel, BigVAR.results, BigVAR.est

Examples

data(Y)
# Fit a Basic VARX-L with rolling cross validation
Model1=constructModel(Y,p=4,struct="Basic",gran=c(50,10))
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,.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)
```

---

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.
plot.BigVAR  

Plot a BigVAR object

Description

Plot a BigVAR object

Usage

## 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

cv.BigVAR

predict

Forecast using a BigVAR.results object

Description

Forecast using a BigVAR.results object

Usage

predict(object,...)

Arguments

object BigVAR.results object from cv.BigVAR
... additional arguments affecting the predictions produced (e.g. n.ahead)
Details

Provides n.ahead step forecasts using the model produced by cv.BigVAR.

See Also

cv.BigVAR

Examples

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)

Description

Default show method for an object of class BigVAR.results

Usage

## 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

```r
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

```r
constructModel
```

---

SparsityPlot.BigVAR.results

Sparsity Plot of a BigVAR.results object

Description

Sparsity Plot of a BigVAR.results object

Usage

```r
SparsityPlot.BigVAR.results(object)
```
**VarptoVar1MC**

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

### 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

```r
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))
```

### Description

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

### Usage

```r
VarptoVar1MC(B, p, k)
```

### 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 Ai as a VAR(1).
VARXFit

References

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

See Also

MultVarSim

Examples

k=3;p=6
B=matrix(0,nrow=k,ncol=p*k)
A1<- matrix(c(.4,.02,.01,-.02,.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)

VARXFit  Fit a VAR or VARX model by least squares

Description

Fit a VAR or VARX model by least squares

Usage

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

Arguments

Y  a t x 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 Schnei-
der (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

References


See Also

constructModel, cv.BigVAR

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)

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
VARXForecastEval

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)

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 over the evaluation period.

References


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,T1,T2,"BIC",1)
Simulated Multivariate Time Series

Description

Realization of a simulated multivariate time series

Details

100 \times 3 \text{ multivariate time series distributed according to the generator matrix } A.

Author(s)

Will Nicholson
Index

A, 2, 18

BigVAR, 2, 3
BigVAR-class, 3
BigVAR-package (BigVAR), 2
BigVAR.est, 4, 8, 9
BigVAR.est,BigVAR-method (BigVAR.est), 4
BigVAR.results, 3, 5, 9, 12, 14
BigVAR.results-class (BigVAR.results), 5

constructModel, 3–5, 6, 9, 11, 13, 15–17
cv.BigVAR, 3, 5, 8, 12, 14, 16, 17
cv.BigVAR,BigVAR-method (cv.BigVAR), 8

MultVarSim, 9, 15

plot, 3, 10
plot,BigVAR-method (plot.BigVAR), 11
plot,BigVAR.results-method (plot), 10
plot.BigVAR, 3, 11
predict, 3, 11
predict,BigVAR.results-method
 (predict), 11

show, 12
show,BigVAR-method (show.BigVAR), 13
show,BigVAR.results-method (show), 12
show.BigVAR, 13
SparsityPlot.BigVAR.results, 13
SparsityPlot.BigVAR.results,BigVAR.results-method
 (SparsityPlot.BigVAR.results), 13

VarptoVar1MC, 9, 10, 14
VARXFit, 15, 17
VARXForecastEval, 16

Y, 18