# Package 'BigVAR'

July 22, 2018

**Type** Package

**Title** Dimension Reduction Methods for Multivariate Time Series

**Version** 1.0.3

**Author** c(
person(``Will", ``Nicholson", email = ``wbn8@cornell.edu", role = c(``cre",``aut")),
person(``David", ``Matteson", email = ``matteson@cornell.edu", role = ``aut"),
person(``Jacob", ``Bien", email = ``bien@cornell.edu", role = ``aut"))

**Maintainer** Will Nicholson <wbn8@cornell.edu>

**Date** 2018-07-22

**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

**SystemRequirements** C++11

**LinkingTo** Rcpp, RcppArmadillo, RcppEigen

**URL** http://www.github.com/wbnicholson/BigVAR

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-07-22 20:40:13 UTC

## R topics documented:

---

A *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

Gredenhoff, Mikael, and Sune Karlsson. "Lag-length selection in VAR-models using equal and unequal lag-length procedures." Computational Statistics 14.2 (1999): 171-187.

BigVAR *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.

## 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

Lutkepohl "New Introduction to Multivariate Time Series", William B Nicholson, Jacob Bien, and David S Matteson. "High Dimensional Forecasting via Interpretable Vector Autoregression." arXiv preprint arXiv:1412.5250, 2016. William B Nicholson, David S. Matteson, and Jacob Bien (2015), "VARX-L Structured regularization for large vector autoregressions with exogenous variables," arXiv preprint arXiv:1508.07497, 2016.. William B Nicholson, David S. Matteson, and Jacob Bien (2016), "BigVAR: Dimension Reduction Reduction Methods for Multivariate Time Series," http://www.wbnicholson.com/BigVAR.pdf.

## 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

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

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)

tol optimization tolerance

lagselect lag selection indicator

window.size size of rolling window. If set to NULL an expanding window will be used.

## See Also

constructModel

`BigVAR.est` *BigVAR Estimation*

## 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)
```

BigVAR.results *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

`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

## Note

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

## Author(s)

Will Nicholson

---

`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), intercept = TRUE,
  tol = 1e-04, lagselect = FALSE, window.size = 0)
```

## 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: option 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 $Y$ |
| intercept | True or False: option to fit an intercept |
| tol | optimization tolerance (default 1e-4) |
| lagselect | lag selection indicator |
| window.size | size of rolling window. If set to 0 an expanding window will be used. |

@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)
- "BGR" (Bayesian Ridge Regression (cf. Banbura et al))

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 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

William B Nicholson, Jacob Bien, and David S Matteson. "High Dimensional Forecasting via Interpretable Vector Autoregression." arXiv preprint arXiv:1412.5250, 2016. William B. Nicholson, David S. Matteson, Jacob Bien,VARX-L: Structured regularization for large vector autoregressions with exogenous variables, International Journal of Forecasting, Volume 33, Issue 3, 2017, Pages 627-651, William B Nicholson, David S. Matteson, and Jacob Bien (2016), "BigVAR: Tools for Modeling Sparse High-Dimensional Multivariate Time Series" arxiv:1702.07094

Banbura, Marta, Domenico Giannone, and Lucrezia Reichlin. "Large Bayesian vector auto regressions." Journal of Applied Econometrics 25.1 (2010): 71-92.

## 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

## 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

```
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)
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

```
## 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

constructModel

---

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)
```

## PredictVARX *One-step ahead predictions for VARX models*

### Description

One-step ahead predictions for VARX models

### Usage

```
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

```
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

```
## 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

```
## 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)
```

### 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).

### 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,.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

```
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

Neumaier, Arnold, and Tapio Schneider. "Estimation of parameters and eigenmodes of multivariate autoregressive models." ACM Transactions on Mathematical Software (TOMS) 27.1 (2001): 27-57.

## 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 |
| 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 as well as the forecasts over the evaluation period and lag order selected.

## References

Neumaier, Arnold, and Tapio Schneider. "Estimation of parameters and eigenmodes of multivariate autoregressive models." ACM Transactions on Mathematical Software (TOMS) 27.1 (2001): 27-57.

## 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 *Simulated Multivariate Time Series*

## Description

Realization of a simulated multivariate time series

## Details

$100 \times 3$ multivariate time series distributed according to the generator matrix A.

## Author(s)

Will Nicholson

# Index