Package 'multivar'

October 13, 2022

Encoding UTF-8

Title Penalized Estimation of Multiple-Subject Vector Autoregressive
(multi-VAR) Models

Version 1.1.0

Description Functions for simulating, estimating and forecasting stationary Vector Autoregressive (VAR) models for multiple subject data using the penalized multi-VAR framework in Fisher, Kim and Pipiras (2020) <arXiv:2007.05052>.

Depends R (>= 2.10)

Imports methods, stats, utils, MASS, Rcpp (>= 1.0.3), Matrix, ggplot2,
vars, reshape2, glmnet

License GPL (>= 2)

LazyData true

ByteCompile true

RoxygenNote 7.1.1

NeedsCompilation yes

Maintainer Zachary Fisher <fish.zachary@gmail.com>

Repository CRAN

LinkingTo Rcpp,RcppArmadillo

VignetteBuilder knitr, rmarkdown

Suggests knitr, rmarkdown

Author Zachary Fisher [aut, cre],
Younghoon Kim [ctb],
Vladas Pipiras [ctb]

Date/Publication 2022-05-27 23:40:02 UTC

R topics documented:

multivar-package   *Penalized Estimation of Multiple Subject Vector Autoregressive (multi-VAR) Models*

Description

multivar is an R package for simulating, estimating and forecasting stationary Vector Autoregressive (VAR) models for multiple subject data using the penalized multi-VAR framework.

Author(s)

Maintainer: Zachary Fisher <fish.zachary@gmail.com>

Other contributors:

- Younghoon Kim [contributor]
- Vladas Pipiras [contributor]

canonical.multivar   *Canonical VAR Fitting Function for multivar*

Description

Canonical VAR Fitting Function for multivar

Usage

```
canonical.multivar(object)
```

Arguments

object   multivar object built using ConstructModel.

Details

A function to fit a canonical VAR model to each individual dataset.

Value

A list of results.

See Also

[constructModel](),

Examples

```
# example 1 (run)
sim1  <- multivar_sim(
  k = 2,  # individuals
  d = 5,  # number of variables
  n = 20, # number of timepoints
  prop_fill_com = 0.1, # proportion of paths common
  prop_fill_ind = 0.05, # proportion of paths unique
  lb = 0.1,  # lower bound on coefficient magnitude
  ub = 0.5,  # upper bound on coefficient magnitude
  sigma = diag(5) # noise
)

model1 <- constructModel(data = sim1$data, weightest = "ols")
fit1 <- canonical.multivar(model1)
```

---

constructModel   *Construct an object of class multivar*

Description

Construct an object of class multivar

Usage

```
constructModel(
  data = NULL,
  lag = 1,
  horizon = 0,
  t1 = NULL,
  t2 = NULL,
  lambda1 = NULL,
  lambda2 = NULL,
  nlambda1 = 30,
  nlambda2 = 30,
  depth = 1000,
  tol = 1e-04,
  window = 1,
```

```
    standardize = T,
    weightest = "lasso",
    canonical = FALSE,
    threshold = FALSE,
    lassotype = "adaptive",
    intercept = FALSE,
    W = NULL,
    ratios = NULL,
    cv = "blocked",
    nfolds = 10,
    thresh = 0,
    lamadapt = FALSE
)
```

Arguments

data	List. A list (length = k) of T by d multivariate time series
lag	Numeric. The VAR order. Default is 1.
horizon	Numeric. Desired forecast horizon. Default is 1. ZF Note: Should probably be zero.
t1	Numeric. Index of time series in which to start cross validation. If NULL, default is floor(nrow(n)/3) where nk is the time series length for individual k.
t2	Numeric. Index of times series in which to end cross validation. If NULL, default is floor(2*nrow(n)/3) where nk is the time series length for individual k.
lambda1	Matrix. Regularization parameter 1. Default is NULL.
lambda2	Matrix. Regularization parameter 2. Default is NULL.
nlambda1	Numeric. Number of lambda1 values to search over. Default is 30.
nlambda2	Numeric. Number of lambda2 values to search over. Default is 30.
depth	Numeric. Depth of grid construction. Default is 1000.
tol	Numeric. Optimization tolerance (default 1e-4).
window	Numeric. Size of rolling window.
standardize	Logical. Default is true. Whether to standardize the individual data.
weightest	Character. Default is "mlr" for multiple linear regression. "sls" for simple linear regression also available. How to estimate the first-stage weights.
canonical	Logical. Default is false. If true, individual datasets are fit to a VAR(1) model.
threshold	Logical. Default is false. If true, and canonical is true, individual transition matrices are thresholded based on significance.
lassotype	Character. Default is "adaptive". Choices are "standard" or "adaptive" lasso.
intercept	Logical. Default is FALSE.
W	Matrix. Default is NULL.
ratios	Numeric vector. Default is NULL.

cv	Character. Default is "rolling" for rolling window cross-validation. "blocked" is also available for blocked folds cross-validation. If "blocked" is selected the nfolds argument should bbe specified.
nfolds	Numeric. The number of folds for use with "blocked" cross-validation.
thresh	Numeric. Post-estimation threshold for setting the individual-level coefficients to zero if their absolute value is smaller than the value provided. Default is zero.
lamadapt	Logical. Should the lambdas be calculated adaptively. Default is FALSE.

Examples

```
sim  <- multivar_sim(
  k = 2,  # individuals
  d = 3,  # number of variables
  n = 20, # number of timepoints
  prop_fill_com = 0.1, # proportion of paths common
  prop_fill_ind = 0.1, # proportion of paths unique
  lb = 0.1,  # lower bound on coefficient magnitude
  ub = 0.9,  # upper bound on coefficient magnitude
  sigma = diag(3) # noise
)

plot_sim(sim, plot_type = "common")

model <- constructModel(data = sim$data, weightest = "ols")
```

---

cv.multivar	*Cross Validation for multivar*

Description

Cross Validation for multivar

Usage

```
cv.multivar(object)
```

Arguments

object	multivar object built using ConstructModel.

Details

The main function of the multivar package. Performs cross validation to select penalty parameters over a training sample and evaluates them over a test set.

Value

An object of class `multivar.results`.

Examples

```
# example 1 (run)
sim1  <- multivar_sim(
  k = 2,  # individuals
  d = 5,  # number of variables
  n = 20, # number of timepoints
  prop_fill_com = 0.1, # proportion of paths common
  prop_fill_ind = 0.05, # proportion of paths unique
  lb = 0.1,  # lower bound on coefficient magnitude
  ub = 0.5,  # upper bound on coefficient magnitude
  sigma = diag(5) # noise
)

model1 <- constructModel(data = sim1$data)
fit1 <- multivar::cv.multivar(model1)
```

---

dat_multivar_sim   *Simulated multi-VAR data.*

Description

This dataset contains multivariate time series data for k = 9 individuals with $d = 10$ variables collected at $t = 100$ equidistant time points. The data was generated such that each individual's VAR(1) transition matrix has 20 percent nonzero entries. This means, for example, each individual has 20 nonzero directed relationships in their data generating model. The position of non-zero elements in each individual's transition matrix was selected randomly given the following constraints: 2/3 of each individual's paths are shared by all individuals, and 1/3 are unique to each individual. For each individual, coefficient values between U(0,1, 0.9) were randomly drawn until stability conditions for the VAR model were satisfied.

Usage

```
dat_multivar_sim
```

Format

A list containing

mat_com a common effects transition matrix

mat_ind_unique a list of unique (individual-specific) effect matrices

mat_ind_final a list of total (common + individual-specific) effect matrices

data a list of multivariate time series for all subjects ...

---

multivar-class   *multivar object class*

Description

An object class to be used with cv.multivar

Details

To construct an object of class multivar, use the function `constructModel`

Slots

k Numeric. The number of subjects (or groupings) in the dataset.

n Numeric Vector. Vector containing the number of timepoints for each dataset.

d Numeric Vector. Vector containing the number of variables for each dataset.

Ak List. A list (length = k) of lagged (T-lag-horizon) by d multivariate time series.

bk List. A list (length = k) of (T-lag-horizon) by d multivariate time series.

Hk List. A list (length = k) of (horizon) by d multivariate time series.

A Matrix. A matrix containing the lagged ((T-lag-horizon)k) by (d+dk) multivariate time series.

b Matrix. A matrix containing the non-lagged ((T-lag-horizon)k) by (d) multivariate time series.

H Matrix. A matrix containing the non-lagged (horizon k) by d multivariate time series.

lag Numeric. The VAR order. Currently only lag 1 is supported.

horizon Numeric. Forecast horizon.

t1 Numeric vector. Index of time series in which to start cross validation for individual k.

t2 Numeric vector. Index of time series in which to end cross validation for individual k.

lambda1 Numeric vector. Regularization parameter 1.

lambda2 Numeric vector. Regularization parameter 2.

nlambda1 Numeric. Number of lambda1 values to search over. Default is 30.

nlambda2 Numeric. Number of lambda2 values to search over. Default is 30.

tol Numeric. Convergence tolerance.

depth Numeric. Depth of grid construction. Default is 1000.

window Numeric. Size of rolling window.

standardize Logical. Default is true. Whether to standardize the individual data.

weightest Character. How to estimate the first-stage weights. Default is "lasso". Other options
include "ridge", "ols" and "var".

canonical Logical. Default is false. If true, individual datasets are fit to a VAR(1) model.

threshold Logical. Default is false. If true, and canonical is true, individual transition matrices
are thresholded based on significance.

lassotype Character. Default is "adaptive". Choices are "standard" or "adaptive" lasso.

intercept Logical. Default is FALSE.

W Matrix. Default is NULL.

ratios Numeric vector. Default is NULL.

cv Character. Default is "blocked" for k-folds blocked cross-validation. rolling window cross-validation also available using "rolling". If "blocked" is selected the nfolds argument should be specified.

nfolds Numeric. The number of folds for use with "blocked" cross-validation.

thresh Numeric. Post-estimation threshold for setting the individual-level coefficients to zero if their absolute value is smaller than the value provided. Default is zero.

lamadapt Logical. Should the lambdas be calculated adaptively. Default is FALSE.

See Also

constructModel

---

multivar_sim   *Simulate multivar data.*

Description

Simulate multivar data.

Usage

```
multivar_sim(
  k,
  d,
  n,
  prop_fill_com,
  prop_fill_ind,
  lb,
  ub,
  sigma,
  unique_overlap = FALSE,
  mat_common = NULL,
  mat_unique = NULL,
  mat_total = NULL,
  diag = FALSE
)
```

Arguments

`k`	Integer. The number of individuals (or datasets) to be generated.
`d`	Integer. The number of variables per dataset. For now this will be constant across individuals.
`n`	Integer. The time series length.
`prop_fill_com`	Numeric. The proportion of nonzero paths in the common transition matrix.
`prop_fill_ind`	Numeric. The proportion of nonzero unique (not in the common transition matrix or transition matrix of other individuals) paths in each individual transition matrix.
`lb`	Numeric. The upper bound for individual elements of the transition matrices.
`ub`	Numeric. The lower bound for individual elements of the transition matrices.
`sigma`	Matrix. The (population) innovation covariance matrix.
`unique_overlap`	Logical. Default is FALSE. Whether the unique portion should be completely unique (no overlap) or randomly chosen.
`mat_common`	Matrix. A common effects transition matrix (if known).
`mat_unique`	List. A list of unique effects transition matrix (if known).
`mat_total`	List. A list of total effects transition matrix (if known).
`diag`	Logical. Default is FALSE. Should diagonal elements be filled first for common elements.

Examples

```
k <- 3
d <- 10
n <- 20
prop_fill_com <- .1
prop_fill_ind <- .05
lb <- 0.1
ub <- 0.5
sigma <- diag(d)
data <- multivar_sim(k, d, n, prop_fill_com, prop_fill_ind, lb, ub,sigma)$data
```

---

plot_results	*Plot data arising from cv.multivar.*

Description

Plot data arising from cv.multivar.

Usage

```
plot_results(
  x,
  plot_type = "common",
  facet_ncol = 3,
  datasets = "all",
  ub = 1,
  lb = -1
)
```

Arguments

x	Object. An object returned by multivar_sim.
plot_type	Character. User can specify "common" to plot the common effects matrix, "unique" to plot the unique effects matrix, or "total" to plot the total effects matrix.
facet_ncol	Numeric. Number of columns to use in the "unique" or "total" effects plot.
datasets	Numeric. A vector containing the index of datasets to plot. Default is "all".
ub	Numeric. Upper bound on coefficient values for heatmap index. Default is 1.
lb	Numeric. Lower bound on coefficient values for heatmap index. Default is -1.

Examples

```
sim1  <- multivar_sim(
  k = 2,  # individuals
  d = 3,  # number of variables
  n = 20, # number of timepoints
  prop_fill_com = 0.1, # proportion of paths common
  prop_fill_ind = 0.1, # proportion of paths unique
  lb = 0.1,  # lower bound on coefficient magnitude
  ub = 0.9,  # upper bound on coefficient magnitude
  sigma = diag(1,3) # noise
)

model1 <- constructModel(data = sim1$data, weightest = "ols")
fit1 <- cv.multivar(model1)
plot_results(fit1, plot_type = "common")
```

---

plot_sim   *Plot data arising from multivar_sim.*

Description

Plot data arising from multivar_sim.

Usage

```
plot_sim(
    x,
    plot_type = "common",
    facet_ncol = 3,
    datasets = "all",
    ub = 1,
    lb = -1
)
```

Arguments

x	Object. An object returned by multivar_sim.
plot_type	Character. User can specify "common" to plot the common effects matrix, "unique" to plot the unique effects matrix, or "total" to plot the total effects matrix.
facet_ncol	Numeric. Number of columns to use in the "unique" or "total" effects plot.
datasets	Numeric. A vector containing the index of datasets to plot. Default is "all".
ub	Numeric. Upper bound on coefficient values for heatmap index. Default is 1.
lb	Numeric. Lower bound on coefficient values for heatmap index. Default is -1.

Examples

```
k <- 3
d <- 5
n <- 50
prop_fill_com <- .2
prop_fill_ind <- .2
lb <- 0.1
ub <- 0.7
sigma <- diag(0.1,d)
sim <- multivar_sim(k, d, n, prop_fill_com, prop_fill_ind, lb, ub,sigma)
plot_sim(sim, plot_type = "common")
```

---

plot_transition_mat *Plot arbitrary transition matrix.*

Description

Plot arbitrary transition matrix.

Usage

```
plot_transition_mat(x, title = NULL, subtitle = NULL, ub = 1, lb = -1)
```

Arguments

x	Matrix. An arbitrary transition matrix.
title	Character. A title for the plot.
subtitle	Character. A subtitle for the plot.
ub	Numeric. Upper bound on coefficient values for heatmap index. Default is 1.
lb	Numeric. Lower bound on coefficient values for heatmap index. Default is -1.

Examples

```
plot_transition_mat(matrix(rnorm(25),5,5), title= "Example")
```

---

show.multivar	*Default show method for an object of class multivar*

Description

Default show method for an object of class multivar

Usage

```
## S4 method for signature 'multivar'
show(object)
```

Arguments

object	multivar object created from ConstructModel

Value

Displays the following information about the multivar object:

• To do.

See Also

constructModel

Index