Package ‘StVAR’

February 11, 2017

Type Package

Title Student's t Vector Autoregression (StVAR)

Version 1.1

Date 2017-02-10

Author Niraj Poudyal

Maintainer Niraj Poudyal <nirajp6@vt.edu>

Description Estimation of
multivariate Student's t dynamic regression models for a given degrees of freedom and lag length. Users can also specify the trends and dummies of any kind in matrix form.

Imports ADGofTest,numDeriv, MCMCpack, matlab

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-11 17:43:11

R topics documented:

StAR *Student’s t Autoregression (StAR)*

Description

Maximum likelihood estimation of StAR model is the purpose of this function. It can be used to estimate the linear autoregressive function (conditional mean) and the quadratic autosckedastic function (conditional variance). Users can specify the model with deterministic variables such as trends and dummies in matrix form.

Usage

```
StAR(Data, Trend=1, lag=1, v=1, maxiter=1000, meth="BFGS", hes="FALSE", init="na")
```

Arguments

`Data`	A data vector with one column. Cannot be empty.
`Trend`	A matrix with columns representing deterministic variables like trends and dummies. If 1 (default), model with only constant intercept is estimated. If 0, the model is estimated without an intercept term.
`lag`	A positive integer (default value is 1) as lag length.
`v`	A scalar (default value is 1) greater than or equal to 1. Degrees of freedom parameter.
`maxiter`	Maximum number of iteration. Must be an integer bigger than 10.
`meth`	One of the optimization method from `optim` function (default value is BFGS). See details of `optim` function.
`hes`	Logical (default value is FALSE). If TRUE produces estimated hessian matrix and the standard errors of estimates.
`init`	If na (default), initial values for optimization are generated from a uniform distribution. A vector of initial values can also be used (not recommended). The length of the init vector must be equal to the number of parameters of the joint distribution.

Details

For the functional form of the autoregressive function and the autoskedastic function, see Spanos (1994) and Poudyal (2012).

Value

`beta`	coefficients of the autoregressive function including the coefficients of trends in matrix form.
`coef`	coefficients of the autoregressive function, standard errors and p-values if `hes=TRUE`. If some of the standard errors are NA's, the `StVAR()` function has to be run again.
`var.coef`	coefficients of the autoregressive function, standard errors and p-values if `hes=TRUE`.
`like`	maximum log likelihood value.
`sigma`	contemporary variance covariance matrix.
`cvar`	`(v/(v+lag*l-2))*sigma*cvar` is the fitted value of the autoskedastic function where `l` is the rank of `Data`
`trend`	estimated trend in the variables.
`res`	nonstandardized residuals
`fitted`	fitted values of the autoregressive function.
`init`	estimates of the joint distribution parameters. It can be used as new initial value `init` in `StVAR()` to improve optimization further.

hes	estimated hessian matrix.
S	variance covariance matrix of the joint distribution.
ad	Anderson-Darling test for Student's t distribution.

Author(s)

Niraj Poudyal <nirajp6@vt.edu>

References

Poudyal, N. (2012), Confronting Theory with Data: the Case of DSGE Modeling. Doctoral dissertation, Virginia Tech.

Spanos, A. (1994), On Modeling Heteroskedasticity: the Student's t and Elliptical Linear Regression Models. Econometric Theory, 10: 286-315.

Examples

```
## StAR Model#####
## Random number seed
set.seed(4093)

## Creating trend variable.
t <- seq(1,100,1)

# Generating data on y and x.
y <-  0.004 + 0.0045*t - 0.09*t^2 + 0.001*t^3 + 50*rt(100,df=5)

# The trend matrix
Trend <- cbind(1,poly(t,3,raw=TRUE))

# Estimating the model
star <- StAR(y,lag=1,Trend=Trend,v=5,maxiter=2000)

# Generate arbitrary dates
dates <- seq(as.Date("2014/1/1"), as.Date("2016/1/1"), "weeks")

## Plotting the variable y, its estimated trend and the fitted value.
d <- dates[2:length(y)] ; Y <- cbind(y[2:length(y)],star$fitted,star$trend)
color <- c("black","blue","black") ; legend <- c("data","trend","fitted values")
cvar <- cbind(star$cvar)
par(mfcol=c(3,1))
matplot(d,Y,xlab="Months",type='l',lty=c(1,2,3),lwd=c(1,1,3),col=color,ylab="",xaxt="n")
axis.Date(1,at=seq(as.Date("2014/1/1"), as.Date("2016/1/1"),"months"),labels=TRUE)
legend("bottomleft",legend=legend,lty=c(1,2,3),lwd=c(1,1,3), col=color,cex=.85)
hist(star$res,main="Residuals",xlab="",ylab="frequency") ## Histogram of y
matplot(d,cvar,xlab="Months",type='l',lty=2,lwd=1,ylab="fitted variance",xaxt="n")
axis.Date(1,at=seq(as.Date("2014/1/1"), as.Date("2016/1/1"),"months"),labels=TRUE)
```

StDLRM *Student's t Dynamic Linear Regression Model (StDLRM)*

Description

Maximum likelihood estimation of StDLRM model is the purpose of this function. It can be used to estimate the dynamc linear autoregressive function (conditional mean) and the quadratic autosckedastic function (conditional variance). Users can specify the model with deterministic variables such as trends and dummies in the matrix form.

Usage

```
StDLRM(y, X, Trend=1, lag=1, v=1, maxiter=1000, meth="BFGS", hes="FALSE", init="na")
```

Arguments

`y`	A vector representing dependent variable. Cannot be empty.
`X`	A data matrix whose columns represent exogeneous variables. Cannot be empty.
`Trend`	A matrix with columns representing deterministic variables like trends and dummies. If 1 (default), model with only constant intercept is estimated. If 0, the model is estimated without an intercept term.
`lag`	A positive integer (default value is 1) as lag length.
`v`	A scalar (default value is 1) greater than or equal to 1. Degrees of freedom parameter.
`maxiter`	Maximum number of iteration. Must be an integer bigger than 10.
`meth`	One of the optimization method from `optim` function (default value is BFGS). See details of `optim` function.
`hes`	Logical (default value is FALSE). If TRUE produces estimated hessian matrix and the standard errors of estimates.
`init`	If na (default), initial values for optimization are generated from a uniform distribution. A vector of initial values can also be used (not recommended). The length of the init vector must be equal to the number of parameters of the joint distribution.

Details

For the functional form of the autoregressive function and the autoskedastic function, see Spanos (1994) and Poudyal (2012).

Value

`beta`	coefficients of the autoregressive function including the coefficients of trends in matrix form.
`coef`	coefficients of the autoregressive function, standard errors and p-values if `hes=TRUE`. If some of the standard errors are NA's, the `StVAR()` function has to be run again.

var.coef	coefficients of the autoskedastic function, standard errors and p-values if if hes=TRUE
like	maximum log likelihood value.
sigma	contemporary variance covariance matrix.
cvar	(v/(v+lag*l-2))*sigma*cvar is the fitted value of the autoskedastic function where l is the rank of Data
trend	estimated trend in the variable y.
res	nonstandardized residuals
fitted	fitted values of the autoregressive function.
init	estimates of the joint distribution parameters. It can be used as new initial value init in StVAR() to improve optimization further.
S	variance covariance matrix of the joint distribution.
ad	Anderson-Darling test for Student's t distribution.

Author(s)

Niraj Poudyal <nirajp6@vt.edu>

References

Poudyal, N. (2012), Confronting Theory with Data: the Case of DSGE Modeling. Doctoral dissertation, Virginia Tech.

Spanos, A. (1994), On Modeling Heteroskedasticity: the Student's t and Elliptical Linear Regression Models. Econometric Theory, 10: 286-315.

Examples

```
## StDLRM Model#####
## Random number seed
set.seed(7504)

## Creating trend variable.
t <- seq(1,100,1)

# Generating data on y and x.
y <-  0.004 + 0.0045*t - 0.09*t^2 + 0.001*t^3 + 50*rt(100,df=5)
x <-  0.05 - 0.005*t + 0.09*t^2 - 0.001*t^3 + 50*rt(100,df=5)
z <-  0.08 - 0.006*t + 0.08*t^2 - 0.001*t^3 + 50*rt(100,df=5)

# The trend matrix
Trend <- cbind(1,poly(t,3,raw=TRUE))

# Estimating the model
stdlrm <- StDLRM(y,cbind(x,z),lag=1,Trend=Trend,v=5,maxiter=2000)

# Generate arbitrary dates
dates <- seq(as.Date("2014/1/1"), as.Date("2016/1/1"), "weeks")

## Plotting the variable y, its estimated trend and the fitted value.
```

```
d <- dates[2:length(y)] ; Y <- cbind(y[2:length(y)],stdlrm$fit,stdlrm$trend)
color <- c("black","blue","black") ; legend <- c("data","trend","fitted values")
cvar <- cbind(stdlrm$cvar)
par(mfcol=c(3,1))
matplot(d,Y,xlab="Months",type='l',lty=c(1,2,3),lwd=c(1,1,3),col=color,ylab=" ",xaxt="n")
axis.Date(1,at=seq(as.Date("2014/1/1"), as.Date("2016/1/1"),"months"),labels=TRUE)
legend("bottomleft",legend=legend,lty=c(1,2,3),lwd=c(1,1,3),col=color,cex=.85)
hist(stdlrm$res,main="Residuals",xlab="",ylab="frequency") ## Histogram of y
matplot(d,cvar,xlab="Months",type='l',lty=2,lwd=1,ylab="fitted variance",xaxt="n")
axis.Date(1,at=seq(as.Date("2014/1/1"),as.Date("2016/1/1"),"months"),labels=TRUE)
```

StVAR *Student's t Vector Autoregression (StVAR)*

Description

Maximum likelihood estimation of StVAR model is the purpose of this function. It can be used to estimate the linear autoregressive function (conditional mean) and the quadratic autosckedastic function (conditional variance). Users can specify the model with deterministic variables such as trends and dummies in the matrix form.

Usage

```
StVAR(Data, Trend=1, lag=1, v=1, maxiter=1000, meth="BFGS", hes="FALSE", init="na")
```

Arguments

`Data`	A data matrix with at least two columns. Cannot be empty.
`Trend`	A matrix with columns representing deterministic variables like trends and dummies. If 1 (default), model with only constant intercept is estimated. If 0, the model is estimated without the intercept term.
`lag`	A positive integer (default value is 1) as lag length.
`v`	A scalar (default value is 1) greater than or equal to 1. Degrees of freedom parameter.
`maxiter`	Maximum number of iteration. Must be an integer bigger than 10.
`meth`	One of the optimization method from `optim` function (default value is BFGS). See details of `optim` function.
`hes`	Logical (default value is FALSE). If TRUE produces estimated hessian matrix and the standard errors of estimates.
`init`	If na (default), initial values for optimization are generated from a uniform distribution. A vector of initial values can also be used (not recommended). The length of the init vector must be equal to the number of parameters of the joint distribution.

Details

For the functional form of the autoregressive function and the autoskedastic function, see Spanos (1994) and Poudyal (2012).

Value

`beta`	coefficients of the autoregressive function including the coefficients of trends in matrix form.
`coef`	coefficients of the autoregressive function, standard errors and p-values. If some of the standard errors are NA's, the `StVAR()` function has to be run again.
`var.coef`	coefficients of the autoskedastic (conditional variance) function, standard errors and p-values.
`like`	maximum log likelihood value.
`sigma`	contemporary variance-covariance matrix.
`cvar`	`(v/(v+lag*l-2))*sigma*cvar` is the fitted value of the autoskedastic function where `l` is the rank of `Data`
`trends`	estimated trends in the variables.
`res`	nonstandardized residuals
`fitted`	fitted values of the autoregressive function.
`init`	estimates of the joint distribution parameters. It can be used as new initial value `init` in `StVAR()` to improve optimization further.
`hes`	the estiamted hessian matrix if `hes=TRUE`.
`S`	variance covariance matrix of the joint distribution.
`ad`	Anderson-Darling test for Student's t distribution.

Author(s)

Niraj Poudyal <nirajp6@vt.edu>

References

Poudyal, N. (2012), Confronting Theory with Data: the Case of DSGE Modeling. Doctoral dissertation, Virginia Tech.

Spanos, A. (1994), On Modeling Heteroskedasticity: the Student's t and Elliptical Linear Regression Models. Econometric Theory, 10: 286-315.

Examples

```
## StVAR Model#####
## Random number seed
set.seed(7504)

## Creating trend variable.
t <- seq(1,100,1)
```

```
# Generating data on y and x.
y <-  0.004 + 0.0045*t - 0.09*t^2 + 0.001*t^3 + 50*rt(100,df=5)
x <-  0.05 - 0.005*t + 0.09*t^2 - 0.001*t^3 + 50*rt(100,df=5)

# The trend matrix
Trend <- cbind(1,poly(t,3,raw=TRUE))

# Estimating the model
stvar <- StVAR(cbind(y,x),lag=1,Trend=Trend,v=5,maxiter=2000)

# Generate arbitrary dates
dates <- seq(as.Date("2014/1/1"), as.Date("2016/1/1"), "weeks")

## Plotting the variable y, its estimated trend and the fitted value.
d <- dates[2:length(y)]; Y <-  cbind(y[2:length(y)],stvar$fit[,1],stvar$trend[,1])
color <- c("black","blue","black") ; legend <- c("data","trend","fitted values")
cvar <- cbind(stvar$cvar)
par(mfcol=c(3,1))
matplot(d,Y,xlab="Months",type='l',lty=c(1,2,3),lwd=c(1,1,3),col=color,ylab=" ",xaxt="n")
axis.Date(1,at=seq(as.Date("2014/1/1"), as.Date("2016/1/1"),"months"),labels=TRUE)
legend("bottomleft",legend=legend,lty=c(1,2,3),lwd=c(1,1,3), col=color,cex=.85)
hist(stvar$res[,1],main="Residuals",xlab="",ylab="frequency") ## Histogram of y
matplot(d,cvar,xlab="Months",type='l',lty=2,lwd=1,ylab="fitted variance",xaxt="n")
axis.Date(1,at=seq(as.Date("2014/1/1"),as.Date("2016/1/1"),"months"),labels=TRUE)
```

Index