Package 'invgamstochvol'

July 22, 2025

Title Obtains the Log Likelihood for an Inverse Gamma Stochastic Volatility Model

Version 1.0.0

Maintainer Blessings Majoni <bmayjay@gmail.com>

Description Computes the log likelihood for an inverse gamma stochastic volatility model using a closed form expression of the likelihood. The details of the computation of this closed form expression are given in Gonzalez and Majoni (2023) <http://rcea.org/RePEc/pdf/wp23-11.pdf> .
The closed form expression is obtained for a stationary inverse gamma stochastic volatility model by marginalising out the volatility. This allows the user to obtain the maximum likelihood estimator for this non linear non Gaussian state space model. In addition, the user can obtain the estimates of the smoothed volatility using the exact smoothing distributions.

License MIT + file LICENSE

Imports Rcpp (>= 1.0.10)

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 2.10)

Suggests knitr, rmarkdown, spelling

LazyData true

NeedsCompilation yes

Encoding UTF-8

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.2.3

Language en-US

Author Leon Gonzalez [aut, cph],
Blessings Majoni [aut, cre] (ORCID:
<https://orcid.org/0009-0006-1793-1241>)

Repository CRAN

Date/Publication 2023-08-18 08:32:42 UTC

Contents

`DrawK0` *Obtains a random draw from the exact posterior of the inverse volatilities.*

Description

Obtains a draw from the posterior distribution of the inverse volatilities.

Usage

```
DrawK0(AllSt, allctil, alogfac, alogfac2, alfac, n, rho, b2, nproc2=2)
```

Arguments

`AllSt`	Some constants obtained from the evaluation of the log likelihood using the function lik_clo
`allctil`	Some constants obtained from the evaluation of the log likelihood using the function lik_clo
`alogfac`	Some constants obtained from the evaluation of the log likelihood using the function lik_clo
`alogfac2`	Some constants obtained from the evaluation of the log likelihood using the function lik_clo
`alfac`	Some constants obtained from the evaluation of the log likelihood using the function lik_clo
`n`	Degrees of freedom.
`rho`	The parameter for the persistence of volatility.
`b2`	Level of volatility.
`nproc2`	The number of processors allocated to the calculations. The default value is set at 2.

Value

A vector with a random draw from the posterior of the inverse volatilities.

Examples

```
##example using US inflation Data
data("US_Inf_Data")
 Ydep <- as.matrix(US_Inf_Data)
 littlerho=0.95
r0=1
rho=diag(r0)*littlerho
p=4
n=4.1
T=nrow(Ydep)
Xdep <- Ydep[p:(T-1),]
if (p>1){
for(lagi in 2:p){
 Xdep <- cbind(Xdep, Ydep[(p-lagi+1):(T-lagi),])
 }
}
T=nrow(Ydep)
Ydep <- as.matrix(Ydep[(p+1):T,])
 T=nrow(Ydep)
unos <- rep(1,T)
Xdep <- cbind(unos, Xdep)

##obtain residuals
 bOLS <- solve(t(Xdep) %*% Xdep) %*% t(Xdep) %*% Ydep
 Res= Ydep- Xdep %*% bOLS
 Res=Res[1:T,1]
 b2=solve(t(Res) %*% Res/T)*(1-rho %*% rho)/(n-2)
 Res=as.matrix(Res,ncol=1)

##obtain log likelihood
   LL1=lik_clo(Res,b2,n,rho)

##obtain smoothed estimates of volatility. First, save the constants from LL1
 deg=200
 niter=200
 AllSt=matrix(unlist(LL1[3]), ncol=1)
 allctil=matrix(unlist(LL1[4]),nrow=T, ncol=(deg+1))
 donde=(niter>deg)*niter+(deg>=niter)*deg
 alogfac=matrix(unlist(LL1[5]),nrow=(deg+1),ncol=(donde+1))
 alogfac2=matrix(unlist(LL1[6]), ncol=1)
 alfac=matrix(unlist(LL1[7]), ncol=1)

 milaK=0
 repli=5
  keep0=matrix(0,nrow=repli, ncol=1)
        for (jj in 1:repli)
       {
          laK=DrawK0(AllSt,allctil,alogfac, alogfac2, alfac, n, rho, b2, nproc2=2)

           milaK=milaK+1/laK*(1/repli)
          keep0[jj]=mean(1/laK)/b2
        }
```

```
        ccc=1/b2
        fefo=(milaK[1:T])*ccc

##moving average of squared residuals
        mRes=matrix(0,nrow=T,ncol=1)
           Res2=Res*Res
         bandi=5
       for (iter in 1:T)
        {  low=(iter-bandi)*(iter>bandi)+1*(iter<=bandi)
          up=(iter+bandi)*(iter<=(T-bandi))+T*(iter>(T-bandi))
          mRes[iter]=mean(Res2[low:up])
        }

##plot the results
       plot(fefo,type="l", col = "red", xlab="Time",ylab="Volatility Means")
          lines(mRes, type="l", col = "blue")
          legend("topright", legend = c("Stochastic Volatility", "Squared Residuals"),
                 col = c("red", "blue"), lty = 1, cex = 0.8)
```

extra *Package to compute the log likelihood for an inverse gamma stochastic volatility model and to draw from the exact posterior of the inverse volatilities.*

Description

This package computes the log likelihood for an inverse gamma stochastic volatility model using a closed form expression of the likelihood.

lik_clo *Compute the log likelihood for an inverse gamma stochastic volatility model*

Description

Computes the log likelihood for an inverse gamma stochastic volatility model using a closed form expression of the likelihood. The details of the computation of this closed form expression are given in Leon-Gonzalez, R., & Majoni, B. (2023). Exact Likelihood for Inverse Gamma Stochastic Volatility Models (No. 23-11). Computations in 'MAC OS' are single-threaded if 'OpenMP' is not installed.

Usage

```
lik_clo( Res,  b2,  n, rho,  NIT=200,  niter=200,  nproc=2,  nproc2=2)
```

Arguments

- `Res` Matrix of OLS residuals. Usually resulting from a call to priorvar.
- `b2` Level of volatility.
- `n` Degrees of freedom.
- `rho` The parameter for the persistence of volatility.
- `NIT` The degree of approximation to truncate the log likelihood sum. The default value is set at 200.
- `niter` The degree of approximation to truncate the hypergeometric sum. The default value is set at 200.
- `nproc` The number of processors allocated to evaluating the hypergeometric function. The default value is set at 2.
- `nproc2` The number of processors allocated to computing the log likelihood. The default value is set at 2.

Details

The closed form expression is obtained for the log likelihood of a stationary inverse gamma stochastic volatility model by marginalising out the volatilities. This allows the user to obtain the maximum likelihood estimator for this non linear non Gaussian state space model. When combined with `DrawK0`, the function can in addition obtain the estimates of the smoothed volatilities using the exact smoothing distributions.

Value

A list of 7 items. List item number 1, is the sum of the log likelihood, while the rest are constants that are useful to obtain the smoothed estimates of the volatility.

Examples

```
##simulate data
n=150
dat<-data.frame(Ydep=runif(n,0.3,1.4))
Ydep <- as.matrix(dat, -1,ncol=ncol(dat))
littlerho=0.95
r0=1
rho=diag(r0)*littlerho
p=4
n=4.1
T=nrow(Ydep)
Xdep <- Ydep[p:(T-1),]
if (p>1){
for(lagi in 2:p){
  Xdep <- cbind(Xdep, Ydep[(p-lagi+1):(T-lagi),])
}
}
T=nrow(Ydep)
 Ydep <- as.matrix(Ydep[(p+1):T,])
 T=nrow(Ydep)
```

```
unos <- rep(1,T)
Xdep <- cbind(unos, Xdep)
##obtain residuals
bOLS <- solve(t(Xdep) %*% Xdep) %*% t(Xdep) %*% Ydep
Res= Ydep- Xdep %*% bOLS
 Res=Res[1:T,1]
 b2=solve(t(Res) %*% Res/T)*(1-rho %*% rho)/(n-2)
 Res=as.matrix(Res,ncol=1)

##obtain log likelihood
LL1=lik_clo(Res,b2,n,rho)
LL1[1]
```

`ourgeo` *Computes the 2F1 Hypergeometric Function*

Description

Computes the 2F1 Hypergeometric Function

Usage

```
ourgeo(a1,a2,b1,zstar,niter=500)
```

Arguments

`a1`	Parameter (Real)
`a2`	Parameter (Real)
`b1`	Parameter (Real)
`zstar`	Primary real argument
`niter`	The degree of approximation to truncate the hypergeometric sum. The default value is set at 500

Value

returns the value of the hypergeometric function

Examples

```
##usage of ourgeo to evaluate a 2F1 hypergeometric function
ourgeo(1.5,1.9,1.2,0.7)
```

`US_Inf_Data` *Data to use in the invgamstochvol package*

Description

Contains the inflation data for the US for the period 1960Q1 to 2022Q3

Usage

```
US_Inf_Data
```

Format

A data frame with 1 variable and 247 observations.

V1 Inflation variable

Source

Federal Reserve Bank of St Louis Fred database as the Consumer Price Index (CPI) data to serve as example data

Examples

```
data(US_Inf_Data) #lazy loading
```

Index