Granger Causality

Code

Srikanth Sarma || ME14B027

04/ 12/ 2017

Causality analysis between different time series has been of interest in many areas of engineering and especially economics. This term project focusses on a popular causality analysis known as the granger causality test.

Question 1:

Hide

```
library(complexplus)
```

The following few are the functions constructed for the purpose of Granger causality testing.

The following function fits a VAR model of given order to the given data. The output is of the form of a list of an array of coefficient matrices and a covariance matrix of the obtained coefficients.

Hide

```
Vhat <- function(Data,Order)
  {
  # Function to obtain V_ij matrix (P*P) from the estimate of covariance matrix (M^
2P*M^2P) of
  # VAR model coefficients
  # Reference: Michael Eichler (2005) and Lutkepohl H (1993)
  # Usage: [Vmat] = Vhat(Data, Order)
  # Sudhakar Kathari, Srikanth Sarma, November 11, 2017.
  # Read input arguments
  Z = Data
  P = Order

  # Size of the data
  Size = dim(Z)
  N = Size[1]
  M = Size[2]

  # The regressor matrix

   BigZ = {}
   BigZ1 = {}

   for (k in 1:P){
     BigZ = cbind(BigZ, Z[k:(N-P+k-1),])
   }

   # Re-arrange the regressor matrix
   for (j in 1:M){
     BigZ1 = cbind(BigZ1,BigZ[,(seq(j,M*P,M))])
   }
   # The Y matrix
   Y = Z[(P+1):N,]

   # Regressors covariance matrix and its inverse
   SigmaR = (t(BigZ1)%*%BigZ1)/N
   invSigmaR = qr.solve(SigmaR)

   # Estimate VAR model coefficient matrices
   Ahat = (qr.solve(t(BigZ1)%*%BigZ1)%*%t(BigZ1))%*%Y

   # Innovations covariance matrix
   SigmaE = (1/(N-P))*t((Y-BigZ1%*%Ahat))%*%(Y-BigZ1%*%Ahat)

   # Covariance matrix of VAR model coefficients
   SigmaA = kronecker(invSigmaR,SigmaE)



   # Estimation of V matrix
  Vmatrix = array(0,dim = c(P,P,M,M))
   for (i in seq(1,M)){
     for(j in seq(1,M))
     Vmatrix[,,i,j] = SigmaA[(((i-1)*M+j-1)*P+1):(((i-1)*M+j)*P),(((i-1)*M+j-1)*P+
```

```
1):(((i-1)*M+j)*P)]
   }
  A<-array(0,dim=c(M,M,P))
  for(i in seq(1,M))
  {
    for(j in seq(1,M))
    {
      for(k in seq(1,P))
      {
        A[i,j,k]<-Ahat[P*(j-1)+k,i]
      }
    }
  }
  c<-list()
  c[[1]]<-A
  c[[2]]<-Vmatrix
   return(c)
}
```

The following function takes in the data and the order of VAR model to be fit and returns a statistic used for the hypothesis testing of the non causality among the time series.

Hide

```
GrangerTest_timeDomain<-function(Zk,P)
{
  c<-Vhat(Zk,P)
  A<-c[[1]]
  V<-c[[2]]
  size=dim(Zk)
  N=size[1]
  M=size[2]

  S<-array(0,dim=c(M,M))
  for(i in seq(1,M))
  {
    for(j in seq(1,M))
    {
      S[i,j]=N*t(A[i,j,])%*%qr.solve(V[,,i,j])%*%A[i,j,]
    }
  }
  return(S)
}
```

The frequency response matrix for a given angular frequency:

Hide

```
FRF_Matrix<-function(A,w)
{
  P=dim(A)[3]
  M=dim(A)[1]
  Abar<-diag(M)
  for(k in seq(1,P))
  {
    Abar<-Abar-A[,,k]*exp(-1i*k*w)
  }

  return(Abar)
}
```

The coherency matrix for a given angular frequency:

Hide

```
coherency_matrix<-function(Abar)
{
  size=dim(Abar)
  M=size[1]

  Psi<-array(0,dim=c(M,M))
  h<-array(0,dim=c(M,M))
  sum<-array(0,dim=M)
  detA<-Det(Abar)

  if(M>3)
  {
    for(i in seq(1,M))
    {
      for(j in seq(1,M))
      {
        if(i==j)
        {
          h[i,j]=Det(Abar[-i,-j])/detA
        }
        else
        {
          h[i,j]=-Abar[i,j]*Det(Abar[-c(i,j),-c(i,j)])/detA
        }
      }
    }
    for(j in seq(1,M))
    {
      for(i in seq(1,M))
      {
        sum[j]=sum[j]+Mod(h[i,j])*Mod(h[i,j])
      }
      sum[j]=sqrt(sum[j])
    }
    for(i in seq(1,M))
    {
      for(j in seq(1,M))
      {
        Psi[i,j]=h[i,j]/sum[i]
      }
    }
  }
  else
  {
    for(i in seq(1,M))
    {
      for(j in seq(1,M))
      {
        if(i==j)
        {
          h[i,j]=Det(Abar[-i,-j])/detA
        }
```

```
        else
        {
          h[i,j]=-Abar[i,j]*Abar[-c(i,j),-c(i,j)]/detA
        }

      }
    }
    for(j in seq(1,M))
    {
      for(i in seq(1,M))
      {
        sum[j]=sum[j]+Mod(h[i,j])*Mod(h[i,j])
      }
      sum[j]=sqrt(sum[j])
    }

    for(i in seq(1,M))
    {
      for(j in seq(1,M))
      {
        Psi[i,j]=h[i,j]/sum[j]
      }
    }
  }

  return(Psi)
}
```

The direct pathway function:

Hide

```
DPF<-function(Zk,P,Nf,plot)
{
  #Nf=N for general fft

  size=dim(Zk)
  N=size[1]
  M=size[2]

  Psi<-array(0,dim=c(M,M,Nf))
  wseq<-seq(0,2*pi,2*pi/(Nf-1))

  c<-Vhat(Zk,P)
  A<-c[[1]]
  V<-c[[2]]

  k=1
  for(w in wseq)
  {
    Abar<-FRF_Matrix(A,w)
    Psi[,,k]<-coherency_matrix(Abar)
    k=k+1
  }

  #Plotting
  if(plot==TRUE)
  {
    par(mfrow=c(M,M))
    for(i in seq(1,M))
    {
      for(j in seq(1,M))
      {
        plot(wseq,Mod(Psi[i,j,])^2,type='l',ylab = expression(Psi^2),xlab = express
ion(omega))
      }
    }
  }

  return(Psi)
}
```

The function to generate surrogate datasets and compute the direct pathway function for each surrogate:

Hide

```
Surrogate_Generation<-function(Zk,R,P,Nf)
{
  Zw<-Mod(mvfft(Zk))
  size=dim(Zk)
  N=size[1]
  M=size[2]
  Zk_surr<-array(0,dim=c(N,M,R))
  Zw_surr<-array(0,dim=dim(Zw))
  Psi_surr<-array(0,dim=c(M,M,Nf,R))
  for(i in seq(1,R))
  {
    for(y in seq(1,M))
    {
      for(x in seq(1,N))
      {
        Zw_surr[x,y]=Zw[x,y]*exp(1i*runif(1,min=0,max=2*pi))
      }
      Zk_surr[,y,i]<-Re(fft(Zw_surr[,y],inverse=TRUE))/length(Zw_surr[,y])
    }
    Psi_surr[,,,i]<-DPF(Zk_surr[,,i],P,Nf,plot = FALSE)
  }

  return(Psi_surr)
}
```

Hide

```
library(MTS)
library(vars)
```

e. Here, we simulate the given series and fit a VAR model to the resulting dataset.

Hide

```
A1<-array(0,dim=c(3,3))
A2<-array(0,dim=c(3,3))
A1[1,1]=0.3
A1[2,1]=0.6
A1[2,2]=0.4
A1[3,2]=0.4
A1[3,3]=0.5
A2[1,1]=0.2
A2[2,2]=0.5
A2[3,2]=0.3
A2[3,3]=0.4
sigma<-diag(3)
z<-VARMAsim(2000,arlags = 2,phi=cbind(A1,A2),sigma = sigma)
z<-z$series
model<-VAR(z,lag.max = 10,ic='SC')
```

```
No column names supplied in y, using: y1, y2, y3 , instead.
```

Hide

```
P<-model$p
```

Time domain granger causality:

Hide

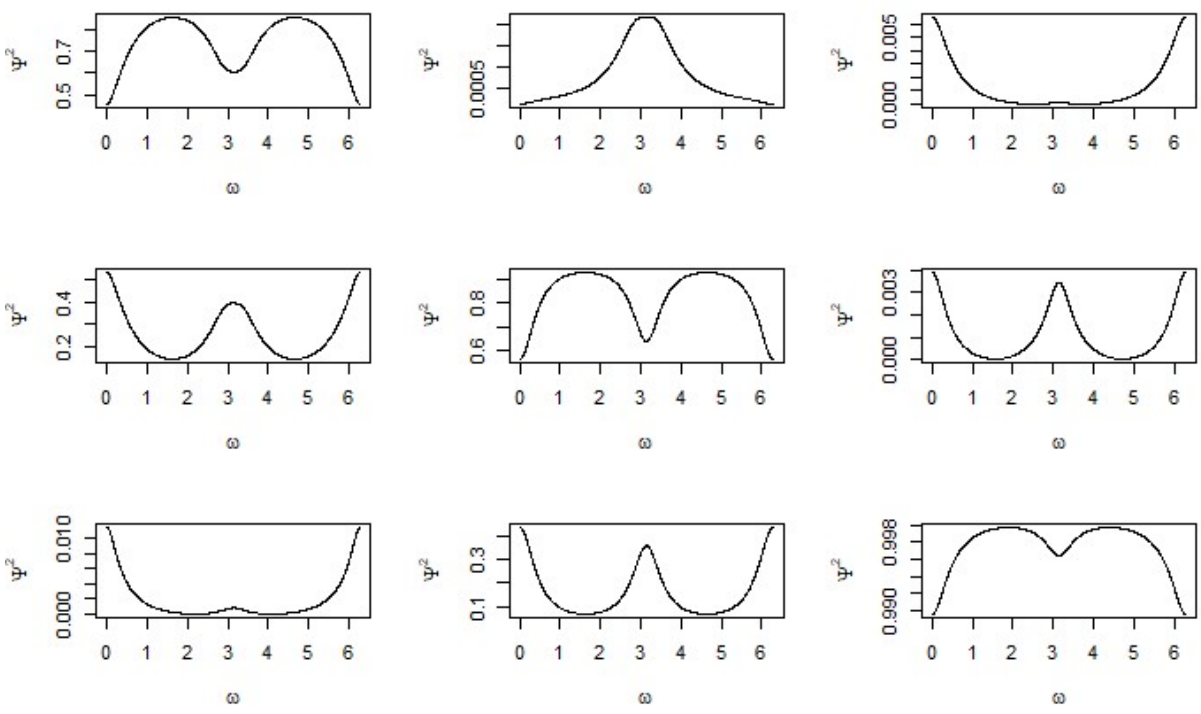
```
GrangerTest_timeDomain(z,P)<qchisq(0.95,df=P)
```

```
      [,1]  [,2]  [,3]
[1,] FALSE  TRUE  TRUE
[2,] FALSE FALSE  TRUE
[3,]  TRUE FALSE FALSE
```

The null hypothesis of non causality cannot be rejected for (1,2), (1,3), (2,3) and (3,1).

Hide

```
Psi<-DPF(z,3,1000,plot = TRUE)
```



From the DPF, we can clearly say that the DPF for (1,2), (1,3), (2,3) and (3,1) are almost zero (very low orders of magnitude). Therefore, neither 2 nor 3 cause 1, 3 doesn't cause 2 and 1 doesn't cause 3.

So, the causality relations as per time domain granger test and visual analysis of DPF are as follows:

- 1 is caused by itself only.
- 2 is caused by 1 and itself.
- 3 is caused by 2 and itself.

Surrogate DPF generation:

Hide

```
R=100
Nf=100
Psi_surr<-Surrogate_Generation(z,R,P,Nf)
```

Frequency domain Granger test:

Hide

```
size=dim(Psi_surr)
M=size[1]
Psi<-DPF(z,3,1000,plot = FALSE)
Psi_95<-array(0,dim=c(M,M,Nf))
non_Cause<-array(TRUE,dim=c(M,M))
for(i in seq(1,M))
{
  for(j in seq(1,M))
  {
    for(k in seq(1,Nf))
    {
      Psi_95[i,j,k]<-quantile(Mod(Psi_surr[i,j,k,])^2,probs=0.95)
      if(Mod(Psi[i,j,k])^2>Psi_95[i,j,k])
      {
        non_Cause[i,j]=FALSE
        break
      }
    }
  }
}
non_Cause
```

```
      [,1]  [,2]  [,3]
[1,]  TRUE  TRUE FALSE
[2,] FALSE  TRUE FALSE
[3,] FALSE FALSE  TRUE
```

The frequency domain results don’t quite agree with the time domain results.

Question 2:

Hide

```
model<-VAR(Zk,lag.max = 10,ic='SC')
```

```
No column names supplied in y, using: y1, y2, y3 , instead.
```

Hide

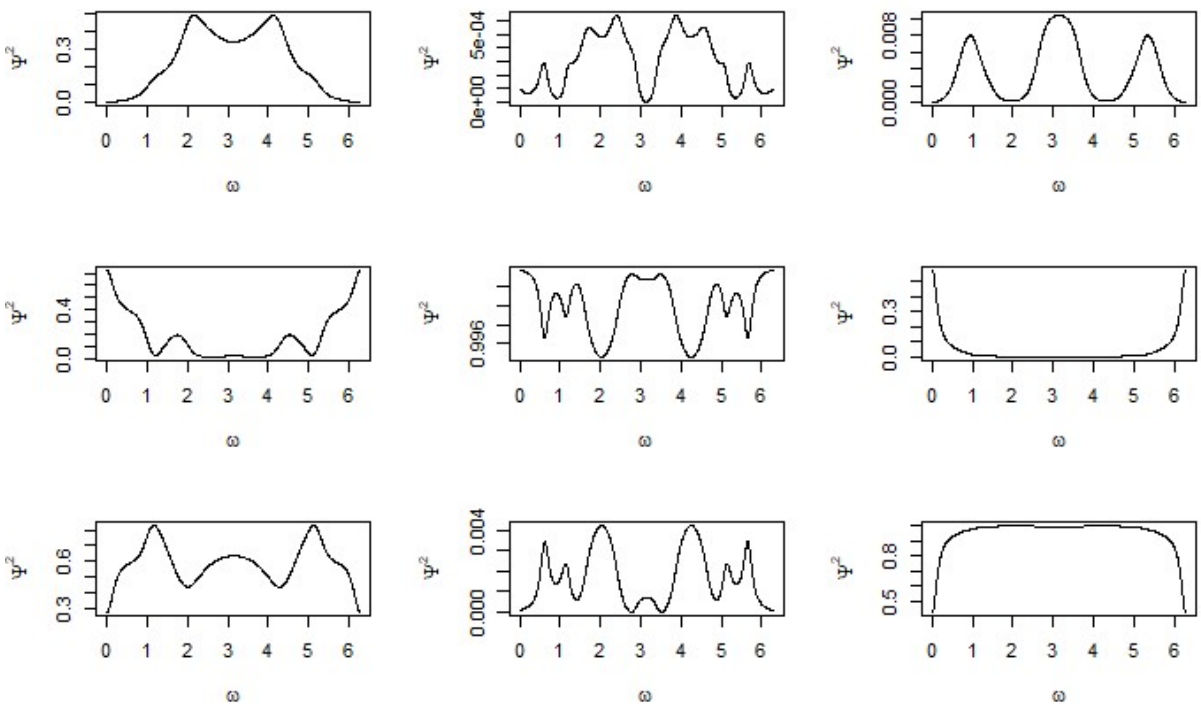
```
P<-model$p
GrangerTest_timeDomain(Zk,P+1)<qchisq(0.95,df=P+1)
```

```
      [,1]  [,2]  [,3]
[1,] FALSE  TRUE  TRUE
[2,] FALSE FALSE FALSE
[3,] FALSE  TRUE FALSE
```

The null hypothesis of non causality cannot be rejected for (1,2), (1,3)and (3,2).

Hide

```
Psi<-DPF(Zk,P+1,1000,plot = TRUE)
```



From the DPF, we can clearly say that the DPF for (1,2), (1,3) and (3,2) are almost zero (very low orders of magnitude). Therefore, neither 2 nor 3 cause 1 and 2 doesn't cause 3.

So, the causality relations as per time domain granger test and visual analysis of DPF are as follows:

- 1 is caused by itself only.
- 2 is caused by 1, 3 and itself.
- 3 is caused by 1 and itself.

Surrogate DPF generation:

Hide

```
R=100
Nf=100
Psi_surr<-Surrogate_Generation(Zk,R,P+1,Nf)
```

Frequency domain Granger causality test:

Hide

```
size=dim(Psi_surr)
M=size[1]
Psi_95<-array(0,dim=c(M,M,Nf))
non_Cause<-array(TRUE,dim=c(M,M))
for(i in seq(1,M))
{
  for(j in seq(1,M))
  {
    for(k in seq(1,Nf))
    {
      Psi_95[i,j,k]<-quantile(Mod(Psi_surr[i,j,k,])^2,probs = 0.95)
      if(Mod(Psi[i,j,k])^2>Psi_95[i,j,k])
      {
        non_Cause[i,j]=FALSE
        break
      }
    }
  }
}
non_Cause
```

```
      [,1]  [,2]  [,3]
[1,]  TRUE FALSE FALSE
[2,] FALSE FALSE FALSE
[3,] FALSE FALSE  TRUE
```

The frequency domain results don’t quite agree with the time domain results.