code2

Lkhagvasuren

6/7/2020

library(tseries)  
library(forecast)  
library(fGarch)  
   
fitting = function(series ,h){  
 model = ar.ols(series,order.max = 4)  
 fcast = predict(model, n.ahead = h)$pred  
 return(fcast) }  
   
# AR(2) -------------------------------------------------------------------  
  
n <- 100   
h <- 12   
  
nsim <- 500   
# ar <- c(.5,.45)   
# ar <- c(.01,.005 )   
ar <- c(.8, .1)   
  
# Boosttrap  
nb <- 100   
# nb <- 20  
# nb <- 60  
  
# Sanamsargui hemjigdehuun  
sd <- 1   
# sd <- 5  
# sd <- 10  
  
b <- round( n^(1/3))   
  
order <- c(2,0,0)   
  
set.seed(3)  
errors\_sim <- matrix(nrow = nsim, ncol = h)  
errors\_bag <- matrix(nrow = nsim, ncol = h)  
  
for(i in 1:nsim){  
 ar.sim <- arima.sim( list(order= order, ar = ar),  
 innov = rt(n+h, df = 40) , # t dist,  
 n = n + h)  
 train <- head( ar.sim, n)  
 test <- tail(ar.sim, h)  
 errors\_sim[i,] <- fitting(train, h) - test  
 errors\_bag[i,] <- colMeans(tsbootstrap(train, nb = nb, b=b, type = 'block',  
 statistic = fitting, h= h)$statistic) - test}  
  
MSE\_ar <- colMeans(errors\_sim^2)  
MSE\_bag <- colMeans(errors\_bag^2)  
  
  
# TAR(2) ------------------------------------------------------------------  
  
### y[t-1] > g == y[t] = a0 + a1 \* y[t-1] + a2 \* y[t-2] + e[t]  
### y[t-1]<= g == y[t] = b0 + b1 \* y[t-1] + b2 \* y[t-2] + e[t]  
  
tar2.sim = function(y0, e , n, p1, p2, th){  
 y = rep(0 ,n)  
 a0 = p1[1]; a1 = p1[2]; a2 = p1[3]   
 b0 = p2[1]; b1 = p2[2]; b2 = p2[3]  
 for(t in 3:n){  
 if(y[t-1] > th) y[t] = a0 + a1 \* y[t-1] + a2 \* y[t-2] + e[t]  
 else y[t] = b0 + b1\* y[t-1] + b2 \* y[t-2] + e[t]}  
 return(y)}  
  
set.seed(3)  
errors\_sim <- matrix(nrow = nsim, ncol = h)  
errors\_bag <- matrix(nrow = nsim, ncol = h)  
  
  
for(i in 1:nsim){  
 tar.sim = tar2.sim(0, e = rt(n+h,df=40),   
 n = n+h,   
 c(0.1,.09,.07),   
 c(0.2 , .05 , .04), th = 0)  
 train <- head( tar.sim, n)  
 test <- tail(tar.sim, h)  
 errors\_sim[i,] <- fitting(train, h) - test  
 errors\_bag[i,] <- colMeans(tsbootstrap(train, nb = nb, b=b, type = 'block',  
 statistic = fitting, h= h)$statistic) - test}  
  
MSE\_ar <- colMeans(errors\_sim^2)  
MSE\_bag <- colMeans(errors\_bag^2)  
  
# GARCH(2,2) --------------------------------------------------------------  
  
spec = garchSpec(model = list(alpha = c(0.2, 0.1),  
 beta = c(0.3,.1)),  
 cond.dist = 'norm')  
  
order <- c(2,0,0)   
  
set.seed(3)  
errors\_sim <- matrix(nrow = nsim, ncol = h)  
errors\_bag <- matrix(nrow = nsim, ncol = h)  
  
for(i in 1:nsim){  
 garch.sim = garchSim(spec, n = n + h, extended = F)$garch  
 train <- head( garch.sim, n)  
 test <- tail(garch.sim, h)  
 errors\_sim[i,] <- fitting(train, h) - test  
 errors\_bag[i,] <- colMeans(tsbootstrap(train, nb = nb, b=n/10, type = 'block',  
 statistic = fitting, h= h)$statistic) - test}  
  
  
MSE\_ar <- colMeans(errors\_sim^2)  
MSE\_bag <- colMeans(errors\_bag^2)