VAR Example MSE Comparison

Deepak Bastola

October 08 2022

```
# install and load the necessary libraries

set.seed(16235, kind = "L'Ecuyer-CMRG" )
pkgs <- c("mvtnorm", "truncnorm", "invgamma","TruncatedNormal", "parallel",
          "Matrix", "ts.extend", "matrixcalc", "mAr", "parallel", "ggthemes")

 if(sum(as.numeric(!pkgs %in% installed.packages())) != 0) {
    installer <- pkgs[!pkgs %in% installed.packages()]
    for(i in 1:length(installer)) {
      install.packages(installer, dependencies = T)
      break()}
    sapply(pkgs, require, character = T)
  } else {
    sapply(pkgs, require, character = T)
  }
```

```
      mvtnorm       truncnorm        invgamma TruncatedNormal        parallel
         TRUE            TRUE            TRUE            TRUE            TRUE
       Matrix       ts.extend      matrixcalc             mAr        parallel
         TRUE            TRUE            TRUE            TRUE            TRUE
     ggthemes
         TRUE
```

```
MSE_comparison <- function(n, p, Sigma, phi, rho, r, c){
  omega = diag(p)
  chain <- as.matrix(mAr.sim(rep(0,p), as.matrix(phi), omega, N = n))

    ar.chain <- lapply(1:p, function(i) ar(chain[,i], order.max = NULL, method = "yw"))
    m <- sapply(1:p, function(i) ar.chain[[i]]$order)
    phi.i <- sapply(1:p, function(i) ar.chain[[i]]$ar)
    sigma.e <- sapply(1:p, function(i) ar.chain[[i]]$var.pred)
    Sigma.pilot <- sapply(1:p, function(i) sigma.e[[i]]/(1 - sum(phi.i[[i]]))^2)

  # Find auto-covariances using function ARMA.autocov() from ts.extend
  ar.autocovar <- lapply(1:p, function(j)
    ARMA.autocov(n = n, ar = phi.i[[j]], ma = 0, corr = FALSE))

  # Current batch size selection method

    ll <- lapply(1:p,
                 function(j) lapply(1:m[[j]],
                                    function(i) sapply(1:i,
                                                       function(k)
```

```
                                                    (k * ar.autocovar[[j]][abs(k-i)+1])))
  ll.sum <- sapply(1:p,
                   function(j) sapply(1:m[[j]],
                                      function(i) sum(ll[[j]][[i]])))
  t1 <- sapply(1:p,
               function(j) sum(phi.i[[j]]*ll.sum[[j]]))
  t2 <- sapply(1:p,
               function(j) ((sigma.e[[j]] - ar.autocovar[[j]][1])/2)*
                 sum(sapply(1:m[[j]],
                            function(i) i*phi.i[[j]])))
  mult <- sapply(1:p,
                 function(i) 1/(1 - sum(phi.i[[i]])))
  gamma.pilot <- -2*(t1 + t2)*mult

# Using function ARMA.autocov() from ts.extend

b <- floor(seq(n^(0.25), n^(0.6),1))
a <- n/b

  gamma0nb <- lapply(1:p,
                     function(j) sapply(1:length(b),
                                        function(i) 2*sum(ar.autocovar[[j]][2:(n-b[i]+1)])))
  gamma0b1 <- lapply(1:p,
                     function(j) sapply(1:length(b),
                                        function(i) 2*sum(ar.autocovar[[j]][2:b[i]])))
  gamma0n1 <- sapply(1:p,
                     function(i) 2*sum(ar.autocovar[[i]][2:n]))
  gamma1b1 <- lapply(1:p,
                     function(j) sapply(1:length(b),
                                        function(i) 2*sum((1:(b[i]-1))*ar.autocovar[[j]][2:b[i]])))
  gamma1n1 <-  sapply(1:p,
                      function(i) 2*sum(ar.autocovar[[i]][2:n]*seq(1, n-1)))
  gamma2b1 <- lapply(1:p,
                     function(j) sapply(1:length(b),
                                        function(i) 2*sum((i^2)*ar.autocovar[[j]][2:b[i]])))


# b/r
br <- floor(b/r)
ar <- n/br

  gamma0b1r <- lapply(1:p,
                      function(j) sapply(1:length(br),
                                         function(i) 2*sum(ar.autocovar[[j]][2:br[i]])))
  gamma0nbr <- lapply(1:p,
                      function(j) sapply(1:length(br), function(i)
                        2*sum(ar.autocovar[[j]][2:(n-br[i]+1)])))
  gamma0n1r <- sapply(1:p, function(i) 2*sum(ar.autocovar[[i]][2:n]))
  gamma1b1r <- lapply(1:p,
                      function(j) sapply(1:length(br),
                                         function(i) 2*sum((1:(br[i]-1))*
                                                             ar.autocovar[[j]][2:br[i]])))
  gamma1n1r <- sapply(1:p, function(i) 2*sum(ar.autocovar[[i]][2:n]*seq(1,n-1)))
```

```
    gamma2b1r <- lapply(1:p,
                        function(j) sapply(1:length(br),
                                           function(i) 2*sum((i^2)*ar.autocovar[[j]][2:br[i]])))


  # AR(p) approximation bias
  # BM

    comb1 <- lapply(1:p,
                    function(j)  -(gamma0n1[[j]] - gamma0b1[[j]] +
                                     gamma1b1[[j]]/b+ b*gamma1n1[[j]]/n^2))
    comb2 <- lapply(1:p,
                    function(j)  -(gamma0n1r[[j]] - gamma0b1r[[j]] +
                                     gamma1b1r[[j]]/br + br*gamma1n1r[[j]]/n^2))

  # multiplier

  mult.bm <- function(n, r, c, b) ((n*r*(c^2 -2*c + r)/((1-c)^2*r*(n*r -b))) -
                                     (b*(c^2 + (1-2*c)*r)/((1-c)^2*r*(n*r-b))))

    bias.arp.bm <- lapply(1:p, function(j)  (1/(1-c))*comb1[[j]] - (c/(1-c))*comb2[[j]])
    var.arp.bm <- lapply(1:p,
                         function(i) 2*Sigma.pilot[i]^2*b/n*
                           ((n*r*(c^2 -2*c + r)/((1-c)^2*r*(n*r -b))) -
                              (b*(c^2 + (1-2*c)*r)/((1-c)^2*r*(n*r-b)))))

    mse.arp.bm <- lapply(1:p, function(i) bias.arp.bm[[i]]^2 + var.arp.bm[[i]])

  # OBM
  expect.arp.obm1 <- lapply(1:p,
                            function(i) gamma0b1[[i]] - gamma0n1[[i]] -
                              (a/(n-b)*gamma1b1[[i]]) + (b*n)/((n-b)*(n-b+1))*
                              (gamma0b1[[i]] - gamma0nb[[i]]))

  expect.arp.obm2 <- lapply(1:p,
                            function(i) gamma0b1r[[i]] - gamma0n1r[[i]] -
                              (ar/(n-br))*gamma1b1r[[i]] +
                              (br*n)/((n-br)*(n-br+1))*(gamma0b1r[[i]] - gamma0nbr[[i]]))

  # multiplier

  mult.obm <- function(n, r, c, b) ((c^2*r - 3*c*r + c + r^2)/((c-1)^2*r^2))

    bias.arp.obm  <- lapply(1:p, function(i) (1/(1-c))*expect.arp.obm1[[i]] -
                              (c/(1-c)*expect.arp.obm2[[i]]))

    variance.arp.obm <- lapply(1:p, function(i) (4/3)*Sigma.pilot[i]^2*
                                 ((c^2*r - 3*c*r + c + r^2)/((c-1)^2*r^2)))
    mse.arp.obm <- lapply(1:p, function(i) bias.arp.obm[[i]]^2 + variance.arp.obm[[i]])

  return(list(b, mse.arp.bm, mse.arp.obm))
}
```

```r
# simulation setup

p <- 4
rho <- 0.92
omega <- diag(p)
A <- matrix(rnorm(p*p,mean=0,sd=1), p, p)
B <- A%*%t(A)
m <- max(eigen(B)$values)
phi0 <- B/(m+0.001)
phi <- bdiag(rho*phi0)

#population covariance

scratch <- diag((p)^2) - kronecker(phi,phi)
V.s <- solve(scratch)%*%vec(diag(p))
V <- matrix(V.s, nrow = p, byrow = TRUE)
Sigma <- solve(diag(p)-phi)%*%V + V%*%solve(diag(p)-phi) -V


# trial
# r1 <- MSE_comparison(n=2e4, p, Sigma, phi, rho, r=1, c=1/2)

# Simulation
# nrep <- 10
# sim12e4 <- mclapply(1:nrep, function(i)
#   MSE_comparison(n= 2e4, p=p, Sigma, phi, rho, r=1, c = 1/2), mc.cores = 4)
# sim32e4 <- mclapply(1:nrep, function(i)
#   MSE_comparison(n= 2e4, p=p, Sigma, phi, rho, r=2, c = 1/2),  mc.cores = 4)
# res <- list(sim12e4, sim32e4)
```