Regularised Cov. Estimation

Jacopo Lussetti

2025-10-27

This report is aiming at replicating the methodology by Basu and Michailidis [2015] on estimating covariance matrices and their precision matrices for high-dimensional data.

Exercise 1

#Data Generation To simulate the stochastic regression model, we will generate data from our code that allows to randomly generate an uppper triangle coefficient matrix. Then we will simulate the data with a VAR(1) model. Then we will proceed to use a Gaussian VAR(1) model \

$$X^t = AX^{t-1} + \epsilon^t, \quad \epsilon^t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\alpha I_{Kp}), \quad \text{where} \quad \alpha = 0.2$$

```r
#we define first of all the various functions
##Simulation upper triangular matrix
### To ensure that process is stable, we need to check the abs of eigenvalue
stab_test <- function(kp, A, tol = 1e-8)
{
  if (!is.matrix(A) || nrow(A) != ncol(A)) {
    stop("The matrix is not square")
  }
  eig <- eigen(A, only.values = TRUE)$values  # computing the eigenvalues

  for (i in 1:length(eig)) {
    if (Mod(eig[i]) >= 1 - tol) {
      return(FALSE)
    }
  }
  return(TRUE)
}

#function to compute companion matrix
comp_mtrx <- function(AA){
    ## AA is a K x Kp matrix, so we are able to derive p in the following way
    K <- nrow(AA)
    Kp <- ncol(AA)
    p <- Kp/K

    # Create the empty companion matrix Kp x Kp
    C <- matrix(0, nrow=Kp, ncol=Kp)

    C[1:K,] <- AA
```

```
    # Add ones on the K-th sub-diagonal
    if (p>1)
        C[(K+1):Kp, 1:(Kp-K)] <- diag(Kp - K)
    return(C)
}
#

## VAR(p) process Simulator
var_sim <- function(AA, nu, Sigma_u, nSteps, y0) {
    K <- nrow(Sigma_u)
    Kp <- ncol(AA)
    p <- Kp/K

    if (p > 1) {
        C <- comp_mtrx(AA) # form the  companion matrix of the var(p) process
    } else {
        C <- AA
    }
    y_t <- matrix(0, nrow =  nSteps, ncol=Kp) #trajectories matrix nSteps x Kp
    y_t[1, 1:Kp] <- y0 #add initial value to initiate the simulation
    noise <- mvrnorm(n = nSteps, mu = rep(0, K), Sigma = Sigma_u) #assuming that
    #residuals follow a multivariate normal distribution

    for (t in 2:nSteps) {
        y_t[t, ] <- C %*% y_t[t-1, ]
        y_t[t, 1:K] <- y_t[t, 1:K] + nu + noise[t,]
    }

    y_t <- zoo(y_t[,1:K], 1:nSteps)
    return(y_t)
  }
```

We will now replicate figure 1 from Basu and Michailidis [2015]. He provided the following values for the spectral radius $\rho(A)$ and fixed diagonal elements of the generating coefficient matrix A to generate the VAR(1) process.

α	$\rho(A)$
0.2	0.2
0.2	0.92
0.2	0.96
0.2	1
0.2	1.01
0.2	1.02
0.2	1.03

To ensure that $\rho(A)$ is fixed to the desired value, we will scale the matrix A by the following scalar:

$$\|A\|_2 = \sqrt{\rho(A^T A)} = a$$
$$\alpha = \sqrt{\frac{a}{\rho(A^T A)}} = \frac{\sqrt{a}}{\sqrt{\rho(A^T A)}}$$

```r
#we first simulate the coef for the predictors
K<-200
A_coef<-matrix(0L,nrow=K, ncol=K)
#we select the upper triangle element
d<-dim(A_coef)
upper_indx<- which(row(A_coef)<col(A_coef)) #index upper triangle element
#we need to ensure that the spectral norm is fixed to a certain value.
#we consider the the first case when 0.2 from figure 1
t<-0
repeat{
  A_coef[upper_indx]<-rnorm(length(upper_indx), mean=0, sd=0.2)
  #we add diagonal values=0.2 as requested from the paper
  diag(A_coef)<-0.2
  #check spectral norm
  spec_norm<-sqrt(max(eigen(t(A_coef)%*%A_coef)$values))
  if (spec_norm-0.2 <1e-2){
    break
  }

  t<-t+1
  if(t>100){
    break
  }
}
#we check also stability
prod<-t(A_coef)%*%A_coef
rho<-max(abs(eigen(prod)$values))
print(sqrt(rho))
```

```
## [1] 4.512884
```

```r
#now we co
nu<-rep(0,K)
cov_res<-diag(0.5, nrow=K, ncol=K)
y_0<-rep(0,K)
#trial  for small n, to then complete the simulation
predictor<-var_sim(AA=A_coef, nu=nu, Sigma_u=cov_res, nSteps=50, y0=y_0)
#now we generate the sparse true coef matrix
# k<-15 # 15 numbers different from zero
# set.seed(1234)
# beta_star<- matrix(0L, nrow=k, ncol=k
```

References

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, pages 1535–1567, 2015.