Package mvtsar

Yuzhe Zhang

2020-12-10

1 Introduction

Package `mvtsar`, whose full name is "matrix-valued time series autoregressive", implements the four algorithms proposed in paper "Autoregressive models for matrix-valued time series". The package is written by complete C++ languague, which relies on two interface package `Rcpp` and `RcppArmadillo`, in order to pursue high computing efficiency. The package contains four algorithms, including VAR, projection method, iterative least squares and maximum likelihood estimate. They will be introduced in the following section.

2 Details

Consider the matrix-valued time series autoregressive model. Specifically, in this model, the conditional mean of the matrix observation at time t is obtained by multiplying the previous observed matrix at time $t-1$ from both left and right by two autoregressive coefficient matrices. Let $\boldsymbol{X}_t$ be the $m \times n$ matrix observed at time t, our model takes the form

$$\boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{X}_{t-1}\boldsymbol{B}' + \boldsymbol{E}_t.$$

We assume that $\mathrm{Cov}(\mathrm{vec}(E_t)) = \Sigma_c \otimes \Sigma_r$, where Σ_r and Σ_c are $m \times m$ and $n \times n$ symmetric positive definite matrices. Σ_r corresponds to row-wise covariances and Σ_c introduces column-wise covariances.

Projection method

The projection method is to solve the following optimization problem

$$\left(\hat{\boldsymbol{A}}_1, \hat{\boldsymbol{B}}_1\right) = \arg\min_{\boldsymbol{A},\boldsymbol{B}} \|\hat{\Phi} - \boldsymbol{B} \otimes \boldsymbol{A}\|_F^2,$$

where $\hat{\Phi}$ is the MLE or LS estimate of model

$$\mathrm{vec}\left(\boldsymbol{X}_t\right) = \Phi\,\mathrm{vec}\left(\boldsymbol{X}_{t-1}\right) + \mathrm{vec}\left(\boldsymbol{E}_t\right).$$

For more details, see the reference paper.

Iterated least squares

The Iterated least squares is to solve the following optimization problem

$$\min_{A,B} \sum_t \|X_t - AX_{t-1}B'\|_F^2\,.$$

To solve it, we iteratively update two matrices $\hat{A}$ and $\hat{B}$

$$\boldsymbol{B} \leftarrow \left(\sum_t \boldsymbol{X}_t'\boldsymbol{A}\boldsymbol{X}_{t-1}\right)\left(\sum_t \boldsymbol{X}_t'\boldsymbol{A}'\boldsymbol{A}\boldsymbol{X}_{t-1}\right)^{-1},$$

$$\boldsymbol{A} \leftarrow \left(\sum_t \boldsymbol{X}_t\boldsymbol{B}\boldsymbol{X}_{t-1}'\right)\left(\sum_t \boldsymbol{X}_{t-1}\boldsymbol{B}'\boldsymbol{B}\boldsymbol{X}_{t-1}'\right)^{-1}.$$

Maximum likelihood estimate

To find the MLE, we iteratively update one, while keeping the other three fixed. These iterations are given by

$$\begin{aligned}
\boldsymbol{A} &\leftarrow \left(\sum_t \boldsymbol{X}_t \Sigma_c^{-1} \boldsymbol{B} \boldsymbol{X}_{t-1}'\right)\left(\sum_t \boldsymbol{X}_{t-1} \boldsymbol{B}' \Sigma_c^{-1} \boldsymbol{B} \boldsymbol{X}_{t-1}'\right)^{-1} \\
\boldsymbol{B} &\leftarrow \left(\sum_t \boldsymbol{X}_t' \Sigma_r^{-1} \boldsymbol{A} \boldsymbol{X}_{t-1}\right)\left(\sum_t \boldsymbol{X}_{t-1}' \boldsymbol{A}' \Sigma_r^{-1} \boldsymbol{A} \boldsymbol{X}_{t-1}\right)^{-1} \\
\Sigma_c &\leftarrow \frac{\sum_t \boldsymbol{R}_t' \Sigma_r^{-1} \boldsymbol{R}_t}{m(T-1)}, \text{ where } \boldsymbol{R}_t = \boldsymbol{X}_t - \boldsymbol{A}\boldsymbol{X}_{t-1}\boldsymbol{B}' \\
\Sigma_r &\leftarrow \frac{\sum_t \boldsymbol{R}_t \Sigma_c^{-1} \boldsymbol{R}_t'}{n(T-1)}, \text{ where } \boldsymbol{R}_t = \boldsymbol{X}_t - \boldsymbol{A}\boldsymbol{X}_{t-1}\boldsymbol{B}'
\end{aligned}.$$

An example

We generate a virtual dataset X, which is of dimension (m, n, t), where t is the number of observations.

```r
library(StatComp20081)
X <- array(1:24, dim = c(2, 3, 4))
VAR(X)
#>             [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
#> [1,]  4.61904762  4.85714286  5.09523810  5.33333333  5.57142857  5.80952381
#> [2,]  3.05476190  3.21428571  3.37380952  3.53333333  3.69285714  3.85238095
#> [3,]  1.49047619  1.57142857  1.65238095  1.73333333  1.81428571  1.89523810
#> [4,] -0.07380952 -0.07142857 -0.06904762 -0.06666667 -0.06428571 -0.06190476
#> [5,] -1.63809524 -1.71428571 -1.79047619 -1.86666667 -1.94285714 -2.01904762
#> [6,] -3.20238095 -3.35714286 -3.51190476 -3.66666667 -3.82142857 -3.97619048
```

```r
PROJ(X)
#> $B.est
#>           [,1]      [,2]      [,3]
#> [1,] -7.614201 -7.972228 -8.330255
#> [2,] -6.392906 -6.694190 -6.995474
#>
#> $A.est
#>            [,1]       [,2]
#> [1,] -10.030826 -11.510322
#> [2,]  -2.006797  -2.336667
#> [3,]   6.017232   6.836988
```

```r
A.init <- matrix(1:4, 2, 2)
B.init <- matrix(1:9, 3, 3)
max.iters <- 200
ILS(X, A.init, B.init, max.iters)
#> $B.est
#>          [,1]     [,2]      [,3]
#> [1,] 715004.6 762.8839 -713379.2
#> [2,] 713810.5 762.9395 -712185.1
#> [3,] 712616.3 762.9950 -710991.0
#>
#> $A.est
#>            [,1]      [,2]
#> [1,] -0.4995833 0.4999999
#> [2,] -0.4999999 0.5004165
```

```r
A.init <- matrix(1:4, 2, 2)
B.init <- matrix(1:9, 3, 3)
sigmar.init <- diag(3)
sigmac.init <- diag(2)
max.iters <- 200
MLE(A.init, B.init, sigmac.init, sigmar.init, X, max.iters)
#> $A.est
#>            [,1]      [,2]
#> [1,] -0.4194144 0.4969825
#> [2,] -0.4969825 0.5745506
#>
#> $B.est
#>            [,1]     [,2]      [,3]
#> [1,] 14.1900079 5.058311 -4.073385
#> [2,]  6.6025408 5.058311  3.514082
#> [3,] -0.9849262 5.058311 11.101549
#>
#> $sigmar.est
#>           [,1]      [,2]
#> [1,] 0.4917674 0.4999313
#> [2,] 0.4999313 0.5082345
#>
#> $sigmac.est
#>              [,1]         [,2]         [,3]
#> [1,] 1.650942e-17 1.721029e-17 1.786696e-17
#> [2,] 1.721029e-17 1.795850e-17 1.865114e-17
#> [3,] 1.786696e-17 1.865114e-17 1.938007e-17
```

Refernces

R. Chen, H. Xiao and D. Yang, Autoregressive models for matrix-valued time series. Journal of Econometrics (2020), https://doi.org/10.1016/j.jeconom.2020.07.015