R Code for Lecture 2

Fu Ouyang

11/23/2018

In what follows, we perform a small-scale Monte Carlo experiment (simulation). First, we draw a time series from the following ARMA(3,2) model (data generating process):

$$X_t = \frac{3}{4}X_{t-1} - \frac{1}{2}X_{t-2} + \frac{1}{4}X_{t-3} + \epsilon_t + \frac{1}{2}\epsilon_{t-1} - \frac{1}{4}\epsilon_{t-2}$$

where $\epsilon_t \overset{iid}{\sim} WN(0, \sigma^2)$. Then we fit an ARMA(3,2) model using the simulated data and the `arima()` function. We draw 1000 random samples for each sample size of 300, 600, and 900. We report mean biases and RMSE of LS and ML estimators for the coefficient on X_{t-1}.

Design 1: Correcly Specified MLE

```
library(TSA)
```

```
simu.1 <- function(TT, S) {

  simu <- function(s) {

    set.seed(s)
    e <- numeric(); X <- numeric()
    e[1:3] <- rnorm(3, 0, 2)
    X[1:3] <- rnorm(3, 0, 1)

    for (t in 4:TT) {
      et <- rnorm(1, 0, 2)
      X[t] = 0.75*X[t-1] - 0.5*X[t-2] + 0.25*X[t-3] + et + 0.5*e[t-1] - 0.25*e[t-2]
      e[t] <- et
    }

    ts.data <- data.frame(X)

    # LSE
    arima.lse <- arima(ts.data, order = c(3, 0, 2), method = "CSS")

    # quasi-Gaussian MLE
    arima.mle <- arima(ts.data, order = c(3, 0, 2), method = "ML")

    c(arima.lse$coef[[1]], arima.mle$coef[[1]])

  }

  simu.results <- t(sapply(1:S, simu))
  table <- matrix(c(colMeans(simu.results - 0.75),
                    sqrt(colMeans((simu.results - 0.75)^2))),
                  nrow = 2, byrow = T)
```

```r
  rownames(table) <- c("bias","RMSE")
  colnames(table) <- c("LS","ML")
  table

}
```

```r
simu.1(300, 1000)
```

```
##               LS         ML
## bias -0.1233900 -0.1000117
## RMSE  0.4108782  0.3580519
```

```r
simu.1(600, 1000)
```

```
##                LS          ML
## bias -0.06872391 -0.05485354
## RMSE  0.29024176  0.26493595
```

```r
simu.1(900, 1000)
```

```
##                LS          ML
## bias -0.04627654 -0.04544759
## RMSE  0.23159659  0.22804454
```

Design 2: Mis-specified MLE

```r
simu.2 <- function(T, S) {

  simu <- function(s) {

    set.seed(s)
    e <- numeric(); X <- numeric()
    e[1:3] <- rgamma(3, 1, 2)
    X[1:3] <- rnorm(3, 0, 1)

    for (t in 4:T) {
      et <- rgamma(1, 1, 2)
      X[t] = 0.75*X[t-1] - 0.5*X[t-2] + 0.25*X[t-3] + et + 0.5*e[t-1] - 0.25*e[t-2]
      e[t] <- et
    }

    ts.data <- data.frame(X)

    # LSE
    arima.lse <- arima(ts.data, order = c(3, 0, 2), method = "CSS")

    # quasi-Gaussian MLE
    arima.mle <- arima(ts.data, order = c(3, 0, 2), method = "ML")

    c(arima.lse$coef[[1]], arima.mle$coef[[1]])

  }
```

```r
  simu.results <- t(sapply(1:S, simu))
  table <- matrix(c(colMeans(simu.results - 0.75),
                    sqrt(colMeans((simu.results - 0.75)^2))),
                  nrow = 2, byrow = T)

  rownames(table) <- c("bias","RMSE")
  colnames(table) <- c("LS","ML")
  table

}
```

```r
simu.2(300, 1000)
```

```
##               LS         ML
## bias -0.05227194 -0.0645972
## RMSE  0.30850291  0.3233850
```

```r
simu.2(600, 1000)
```

```
##               LS          ML
## bias -0.02812475 -0.03557639
## RMSE  0.23994002  0.26100915
```

```r
simu.2(900, 1000)
```

```
##               LS         ML
## bias -0.01957232 -0.0220831
## RMSE  0.20459106  0.2075135
```