Time Series Analysis

Computer Aided Lab 2

This is an ungraded lab sheet. Graded work will appear under Assignments.

For this lab we will perform model identification, parameter estimation and model diagnostics for ARIMA Models. Before we begin load the `lab2` data into `R`. Ensure your working directory is set correctly.

1 Forecast Package

`R` provides many external packages to analyse time series such as `forecast` and `TSA` (the package for the Cryer and Chan textbook). To install the `forecast` package if you have not previously used it on this computer run

```
install.packages("forecast")
```

then to load it run **library(forecast)**

The forecast package provides the function `tsdisplay` for plotting a time series along with the ACF and PACF plots.

2 ARIMA Simulation

You can simulate your own random time series data to experiment on using **arima.sim()** from the TSA package. You have to specify the model as the first argument and the length of the time series as the second. The model is composed of a **list()** and you can specify one or more of **ar** parameters, **ma** parameters, and the (p, d, q) order. For example, to simulate an autoregressive time series of order 1 and length 200 with $\phi = 0.5$ you could use either of

```
x=arima.sim(list(ar=0.5), 200)
x=arima.sim(list(ar=0.5, order=c(1,0,0)), 200)
```

In the first version, the order is automatically inferred from the fact that a single AR parameter was specified, no MA parameters, and no request for $d > 0$. Specifying the order is also useful to detect inconsistent specification (for example if the roots of the AR render it non-stationary).

3 ARMAacf

Try looking at the results of **ARMAacf()**. This function returns the *theoretical* autocorrelations and partial autocorrelations for ARMA models given AR and / or MA parameters.

- **Run the command:**

  ```
  ARMAacf(ar=0.5, ma=0.5, lag.max=5)
  ```

  and code functions to compute γ_0, γ_1, and ρ_1 based on the theory (see 2_ARIMA Tutorial solutions) . Specifically:

$$\gamma_0 = \frac{1 - 2\phi\theta + \theta^2}{1 - \phi^2}\sigma^2$$

$$\gamma_1 = \phi\gamma_0 - \theta\sigma^2$$
$$\rho_1 = \frac{\gamma_1}{\gamma_0}$$

You should obtain the same value of ρ_1, provided you account for the change in sign of θ. **Recall that R uses $+\theta\varepsilon_{t-1}$ whereas we follow Cryer and Chan in writing our equation as $-\theta\varepsilon_{t-1}$.**

- **Also try**

  ```
  ARMAacf(ar=0.5, ma=0.5, lag.max=5, pacf=TRUE)
  ```

  to get the partial autocorrelations.

- **Contrast these with plots based on e.g.**

  ```
  acf(arima.sim(n=1e2,model=list(ar=0.5, ma=0.5)), lag.max=5)
  ```

  Try varying the number of observations n. You can print the ACF values by adding $acf to the end of the command. The sample values converge to the theoretical (expected) values as $n \to \infty$.

4 Parameter Estimation

There are multiple options in R for fitting ARIMA models; **Arima()** from the `forecast` package, **arima()** from the `stats` package, and **arima()** from the `TSA` package. The last two have the same name and are almost identical but the TSA version has some additional functionality. The first argument to **arima()** is the time series data and the second is the order (p, d, q).

- Read in lab2.csv using

  ```
  lab2 = read.csv("lab2.csv")
  ```

- **Fit an ARMA model to Series 7 as follows, where p is the order you identified and d the number of differences you performed.**

  ```
  model7 = arima(lab2$Series7, order=c(p,d,0))
  ```

To perform a significance test on the parameter estimated use **coeftest()** from the `lmtest` package.

```
require(lmtest)
coeftest(model7)
```

Parameters that return non-significant p-values should be considered for removal from the model. Here the mean comes out to not significant. You can fit a model with a mean set to zero using

```
model7=arima(lab2$Series7, order=c(p,d,q),include.mean=FALSE)
```

5 Diagnostics

Recall the 5 steps to perform model diagnostics by examining the residuals of the fitted ARIMA model:

1. Plot the residuals versus time.
2. qq-plot of the residuals.
3. Plot residuals versus the fitted values.
4. ACF plot of residuals and Ljung-Box test.
5. PACF plot of residuals.

You can extract the residuals from a fitted model obtained from **arima()** using the **residuals()** function.

- **Use qqnorm(residuals(model7)) to examine normality of the residuals.**
- **Use qqline(residuals(model7)) to add a (0,1) line.**
- **Use shapiro.test(residuals(model7)) perform a formal hypothesis test for normality.**
- **Use Box.test(residuals(model7),type="Ljung-Box",lag=p+q+10,fitdf=p+q)** to perform the χ^2_{K-p-q} test on the autocorrelations of the residuals. The **lag** argument is K in the notes; it should be set to capture al the non-null ACFs and PACFs, so **at least** $p+q+1$.
  The statistic is the weighted sum of the squared residuals:
  $$Q = n(n+2)\sum_{k=1}^{K} \frac{\left(\hat{\rho}_k^{(\hat{\varepsilon})}\right)^2}{n-k}.$$
  The test is whether the sum of the squared autocorrelations of the residuals are what we'd expect from a white noise.

Practice Exercises:

Do try these before looking at the solutions on the next page!

Exercise 1. Simulate various time series and examine them using **tsdisplay()**.

1. **AR(1) for various ar parameters.**
2. **AR(2) for various ar parameters.**
3. **MA(1) for various ma parameters.**
4. **I(1)**
5. **ARIMA(0,1,1) for various ma parameters.**
6. **ARIMA(1,1,1) for various ar and ma parameters.**

Exercise 2. Model Identification:

1. **Do the simulated series always look like they are a series simulated from the requested model?**

2. **What model do you identify for Series 7?**

Note the series may need to be **differenced** before you can identify the correct ARMA model. Check your notes to help identify how to interpret the order of an ARMA model from the `acf` and `pacf`.

Practice Exercises: Solutions

Exercise 1. Simulate various time series and examine them using **tsdisplay()**.

1. **AR(1) for various ar parameters.**
   y=arima.sim(model=list(ar=0.5,order=c(1,0,0)),n=200)
   tsdisplay(y) # autocorrelated
   y=arima.sim(model=list(ar=0.1,order=c(1,0,0)),n=200)
   tsdisplay(y) # looks like it could be white noise
   y=arima.sim(model=list(ar=0.9,order=c(1,0,0)),n=200)
   tsdisplay(y) # looks like it could be non-stationary; highly autocorrelated

2. **AR(2) for various ar parameters.**
   y=arima.sim(model=list(ar=c(0.5,0.45),order=c(2,0,0)),n=200)
   tsdisplay(y) # looks like it could be non-stationary; highly autocorrelated
   y=arima.sim(model=list(ar=c(0.6,0.5),order=c(2,0,0)),n=200) # doesn't run because non-stationary

3. **MA(1) for various ma parameters.**
   y=arima.sim(model=list(ma=-0.9,order=c(0,0,1)),n=200)
   tsdisplay(y) #

4. **I(1)**
   y=arima.sim(model=list(order=c(0,1,0)),n=200)
   tsdisplay(y) # linear decay of ACFs

5. **ARIMA(0,1,1) for various ma parameters.**
   y=arima.sim(model=list(ma=0.9,order=c(0,1,1)),n=200)
   tsdisplay(y) # linear decay of ACFs
   y=arima.sim(model=list(ma=-1,order=c(0,1,1)),n=200)
   tsdisplay(y) # looks like white noise! This is because we get cancellation of the unit root term on both sides of the ARIMA equation.

6. **ARIMA(1,1,1) for various ar and ma parameters.**
   y=arima.sim(model=list(ar=0.5,ma=0.5,order=c(1,1,1)),n=200)
   tsdisplay(y) #

Exercise 2. Model Identification:

1. **Do the simulated series always look like they are a series simulated from the requested model?**
   No, especially if $\phi \simeq 1$ or some of $\phi_i \simeq 0$, etc. Also, the smaller the time series the harder it is to be sure of (p, d, q)

2. **What model do you identify for Series 7?**
   exponential decay of ACF, first two lags non-null for PACF so AR(2) $\equiv$ ARIMA(2,0,0).