

# Homework 6

Due: Tuesday 3/10/20 by 10:00am

This week, we'll try modeling dependence over time using an moving average model of order  $q$ :

$$y_t = \mu + \sum_{i=1}^q \theta_i w_{t-i} + w_t, \quad w_t \stackrel{i.i.d.}{\sim} \text{normal}(0, \sigma_w^2) \quad (1)$$

Assume  $y_t$  is a stationary process, i.e. assume that the process has constant mean  $\mathbb{E}[y_t] = \mu$ , finite variance  $\mathbb{V}[y_t] = \gamma_y(0)$ , and autocovariances  $\text{Cov}[y_t, y_{t-h}] = \gamma_y(|h|)$ .

## 1. Checking Invertability and Examining the Autocorrelation Function

(a) First, let's explore how to check if values of  $\theta_1, \dots, \theta_q$  correspond to an invertible process.

The R library `polynom` lets us easily compute the roots of polynomials. You'll need to install the `polynom` library and load it. Here's a little example:

```
library(polynom)

# Create a "polynomial" object for the polynomial
# 1 - 5x + 3x^2 + 2x^3
pol <- polynomial(c(1, -5, 3, 2))
# Get the values of x for which 1 - 5x + 3x^2 + 2x^3 = 0
sol <- solve(pol)
```

You may get complex roots  $r = a + bi$ . Note that the absolute value of a complex number  $r$  is given by  $|r| = \sqrt{a^2 + b^2}$ .

Consider the following  $\mathbf{MA}(q)$  models, all with  $\sigma_w^2 = 1$ .

- i.  $q = 1, \theta_1 = 8.11$
- ii.  $q = 2, \theta_1 = -5.64, \theta_2 = -9.34$
- iii.  $q = 2, \theta_1 = 2.15, \theta_2 = 3.69$
- iv.  $q = 3, \theta_1 = 2.08, \theta_2 = 3.39, \theta_3 = 0.73$

(a) For (i)-(iii), find the root of the moving average polynomial that is smallest in magnitude by solving  $\theta(z) = 0$  for  $z$  by hand, without using any special R functions. For (iv), use `polynom` to find the root that is smallest in magnitude. Give the value of this root and indicate whether or not the model is invertible.

(b) Compute the autocorrelation function for (i)-(iv) for values of  $h \geq 0$  by hand.

(c) Using `ARMAacf` to compute  $\rho_x(h)$ , plot the autocorrelation function  $\rho_x(h)$  for  $h = 0, \dots, 10$  for the  $\mathbf{MA}(q)$  models on a single plot. Include a dotted horizontal line at 0.

(d) In general, it isn't straightforward to take a non-invertible  $\mathbf{MA}(q)$  model and transform it to an invertible  $\mathbf{MA}(q)$  model. By straightforward, I mean that there is not simple function that takes non-invertible  $\mathbf{MA}(q)$  parameters and returns the corresponding invertible  $\mathbf{MA}(q)$  that yield the same autocorrelation function. Fortunately, we can use R to do this numerically by taking a set of possibly non-invertible moving average parameters  $\theta_1, \dots, \theta_q$  and:

- Using `ARMAacf` to compute the autocorrelation function  $\rho_y(h)$  for  $h = 0, \dots, h_{max}$  for some  $h_{max}$ ;

- Using `acf2AR` to find the stationary  $AR(h_{max})$  parameters that perfectly characterize the autocorrelation function for  $\rho_y(h)$  for  $h = 0, \dots, h_{max}$  for some  $h_{max}$ ;
- Using `ARMAtoMA` to take the  $AR(h_{max})$  parameters and compute the corresponding *invertible* moving average  $MA(q)$  model (remember - every autoregressive model has a possibly infinite order moving average representation).

To check your work, you can compare the autocorrelation functions  $\rho_y(h)$  for  $h = 0, \dots, h_{max}$  obtained using `ARMAacf` for the possibly non-invertible  $MA(q)$  model, the  $AR(h_{max})$  representations, and the invertible  $MA(q)$  model.

Provide the invertible representations of the non-invertible  $MA(q)$  models in (a).

2. We're going to keep working with residuals from the `broc` data for a little while longer. Again, it is posted on the course website, which contains the average price of one pound of broccoli in urban areas each month, from July 1995 through December 2019. Throughout this problem, we'll continue to work with the residuals from fitting a linear model with a linear time trend and month effects to all but the last 12 months of data. We'll call them  $y$ , because we'll be thinking of them as our observed time series, and we'll define  $m = n - 12$ .

```
load("~/Dropbox/Teaching/TimeSeries2020/stat697/content/data/broc.RData")
set.seed(1)
broc$date <- as.Date(broc$date, "%Y-%m-%d")
broc$month <- format(broc$date, "%m")
broc$dayssincestart <- as.numeric(broc$date) - min(as.numeric(broc$date))
n <- nrow(broc)
m <- nrow(broc) - 12
linmod <- lm(price~dayssincestart+factor(month), data = broc,
             subset = 1:m)
pred <- predict(linmod, broc)
y <- broc$price - pred
```

- (a) Now let's fit some  $MA(q)$  models using `arima` again. We are going to consider the same models that we considered on the previous homework. For  $q = 1, 2, 4, 8, 16, 32$ , compute estimates of  $\mu$ ,  $\theta_1, \dots, \theta_q$  using `arima`. Make a plot with 6 panels. Using one panel for each value of  $q$ , plot the last 24 observations and the estimated fitted values  $\hat{\mathbb{E}}[y_t | y_1, \dots, y_{t-1}]$  from the corresponding fitted model obtained using the Durbin-Levinson algorithm, and the approximate 95% confidence intervals for the estimated fitted values obtained using the parametric bootstrap. You can use `arima.sim` to simulate  $MA(q)$  processes with the mean, moving average parameters, and variance that you estimated from the data. I recommend you construct your 95% intervals by using the parametric bootstrap to approximate  $\mathbb{V}[\hat{\mathbb{E}}[y_t | y_1, \dots, y_{t-1}] | y_1, \dots, y_{t-1}]$  and then assuming that  $\hat{\mathbb{E}}[y_t | y_1, \dots, y_{t-1}]$  is approximately normal - i.e. a  $1 - \alpha$  interval computed in this way would be:

$$\left( \hat{\mathbb{E}}[y_t | y_1, \dots, y_{t-1}] + z_{\alpha/2} \sqrt{\mathbb{V}[\hat{\mathbb{E}}[y_t | y_1, \dots, y_{t-1}] | y_1, \dots, y_{t-1}]}, \right. \\ \left. \hat{\mathbb{E}}[y_t | y_1, \dots, y_{t-1}] + z_{1-\alpha/2} \sqrt{\mathbb{V}[\hat{\mathbb{E}}[y_t | y_1, \dots, y_{t-1}] | y_1, \dots, y_{t-1}]} \right).$$

Code for implementing the Durbin-Levinson algorithm is given below.

```
solve.dl <- function(n, theta = 0, phi = 0, sig.sq.w = 1) {
  C <- matrix(nrow = n, ncol = n)
  v <- rep(NA, n + 1)
  marep <- ARMAtoMA(ar = phi, ma = theta, lag.max = 1000)
  gamma.x <- ARMAacf(ar = phi, ma = theta,
                    lag.max = n)*(1 + sum(marep^2))*sig.sq.w
  gamma.x.0 <- gamma.x[1]
```

```

C[1, 1] <- gamma.x[2]/gamma.x.0
v[1] <- gamma.x.0
v[2] <- v[1]*(1 - C[1, 1]^2)
for (i in 2:n) {
  C[i, i] <- gamma.x[i + 1]
  for (j in 1:(i - 1)) {
    C[i, i] <- C[i, i] - C[i-1, j]*gamma.x[i - j + 1]
  }
  C[i, i] <- C[i, i]/v[i]
  for (j in (i-1):1) {
    C[i, j] <- C[i - 1, j] - C[i, i]*C[i - 1, i - j]
  }
  v[i + 1] <- v[i]*(1 - C[i, i]^2)
}
return(list("C"=C, "c.n" = C[nrow(C), ],
           "v" = v, "v.n" = v[length(v)]))
}

```

- (b) Now for comparison, let's fit some  $AR(p)$  models using `arima` again again. We are going to consider the same models that we considered on the previous homework. For  $p = 1, 2, 4, 8, 16, 32$ , compute estimates of  $\mu, \phi_1, \dots, \phi_p$  using `arima`. Make a plot with 6 panels. Using one panel for each value of  $p$ , plot the last 24 observations and the estimated fitted values  $\hat{\mathbb{E}}[y_t|y_1, \dots, y_{t-1}]$  from the corresponding fitted model obtained using the Durbin-Levinson algorithm, and the approximate 95% confidence intervals for the estimated fitted values obtained using the parametric bootstrap. You can use `arima.sim` to simulate  $AR(p)$  processes with the mean, moving average parameters, and variance that you estimated from the data. I recommend you construct your 95% intervals by using the parametric bootstrap to approximate  $\mathbb{V}[\hat{\mathbb{E}}[y_t|y_1, \dots, y_{t-1}] | y_1, \dots, y_{t-1}]$  and then assuming that  $\hat{\mathbb{E}}[y_t|y_1, \dots, y_{t-1}]$  is approximately normal - i.e. a  $1 - \alpha$  interval computed in this way would be:

$$\left( \hat{\mathbb{E}}[y_t|y_1, \dots, y_{t-1}] + z_{\alpha/2} \sqrt{\mathbb{V}[\hat{\mathbb{E}}[y_t|y_1, \dots, y_{t-1}] | y_1, \dots, y_{t-1}]}, \right. \\ \left. \hat{\mathbb{E}}[y_t|y_1, \dots, y_{t-1}] + z_{1-\alpha/2} \sqrt{\mathbb{V}[\hat{\mathbb{E}}[y_t|y_1, \dots, y_{t-1}] | y_1, \dots, y_{t-1}]} \right).$$

- (c) Plot the autocorrelation function of the first  $m$  observations. If you were considering fitting an  $MA(q)$  model, what would you choose  $q$  to be based on the autocorrelation function alone? Make sure that you choose the maximum value of  $h$  for plotting your autocorrelation function  $\rho_y(h)$  to reflect your choice.
- (d) Plot the partial autocorrelation function of the first  $m$  observations. If you were considering fitting an  $AR(p)$  model, what would you choose  $p$  to be based on the partial autocorrelation function alone? Make sure that you choose the maximum value of  $h$  for plotting your autocorrelation function  $\rho_y(h)$  to reflect your choice.
- (e) Based on (a), (b), (c), (d), and the results of previous analyses on previous assignments, do you think an AR model or an MA model is more appropriate for this data?