Homework 4

Due: Thursday 2/28/19 by 12:00pm (noon)

Although this homework assignment is not due until Thursday 2/28/19, there may be related material on your exam on Friday, 2/22/19.

Forecasting

1. In class, we derived the **forecasting equation** for computing $\hat{x}_{n+1} = \sum_{j=1}^{n} c_{nj} x_{n-j}$ based on $x_1, \dots, x_n$ by minimizing

$$v_n = \mathbb{E}\left[\left(x_{n+1} - \sum_{j=1}^{n} c_{nj} x_{n-j}\right)^2\right].$$

(a) Write down the quantity we should minimize if we want to forecast values further into the future, i.e. if we want to compute $\hat{x}_{n+k}$ for any $k > 0$.

(b) Write out the forecasting equation that correspnods to (a). Hint: In class we talked about minimizing v_n. Defining $a_{n,ij} = \gamma_x(i-j)$ and $b_{n,i} = \gamma_x(i)$, the corresponding forecasting equation was:

$$\boldsymbol{A}_n \boldsymbol{c}_n = \boldsymbol{b}_n.$$

(c) In class, I showed you the following function for computing the values $\boldsymbol{c}_n$ and v_n for a **ARMA**$(1,1)$ model.

```
solve.direct <- function(n, phi1 = 0, theta1 = 0, sig.sq.w = 1) {
  A.n <- matrix(nrow = n, ncol = n)
  b.n <- numeric(n)
  for (i in 1:n) {
    b.n[i] <- gamma.x(i, phi1 = phi1, theta1 = theta1, sig.sq.w = sig.sq.w)
    for (j in 1:n) {
      A.n[i, j] <- gamma.x(h = i - j, phi1 = phi1, theta1 = theta1, sig.sq.w = sig.sq.w)
    }
  }
  c.n <- solve(A.n)%*%b.n
  v.n <- gamma.x(0, phi1 = phi1, theta1 = theta1, sig.sq.w = sig.sq.w) +
    t(c.n)%*%A.n%*%c.n - 2*t(c.n)%*%b.n
  return(list("c.n" = c.n, "v.n" = v.n))
}
```

Modify this function to take an additional argument, k, and return the coefficients that give the forecast $\hat{x}_{n+k}$ as well as the expected squared error loss of $\hat{x}_{n+k}$. Note that you will need to use the `gamma.x` function from class.

```
gamma.x <- function(h, phi1, theta1, sig.sq.w) {
  h <- abs(h)
  if (h == 0) {
    g.x <- (theta1^2 + 2*phi1*theta1 + 1)*sig.sq.w/(1 - phi1^2)
  } else {
    g.x <- sig.sq.w*phi1^(h - 1)*((1 + theta1*phi1)*(phi1 + theta1)/(1 - phi1^2))
  }
  return(g.x)
}
```

(d) Use the function you wrote in (c) to plot the expected squared error loss of $\hat{x}_{3+k}$ for the following models for $k = 1, \dots, 5$, all with $\sigma_w^2 = 2$:

i. $\phi_1 = 0.5$, $\theta_1 = 0$;;
ii. $\phi_1 = -0.5$, $\theta_1 = 0$;;
iii. $\phi_1 = 0$, $\theta_1 = 0.57735$;
iv. $\phi_1 = 0$, $\theta_1 = -0.57735$.

Include a horizontal dashed line at $\gamma_x(0)$ (Note: $\gamma_x(0)$ is the same for all of the models).

(e) In one sentence, interpret how the forecast error relates to the sign of ϕ_1 or θ_1.

(f) In one sentence, interpret how the forecast error decays as k increases for an $\mathbf{AR}(1)$ model versus an $\mathbf{MA}(1)$ model.

(g) Add two more lines to your plot in (c) corresponding to the following models with $\sigma_w^2 = 0.2533333$:

v. $\phi_1 = 0.9$, $\theta_1 = 0$;

vi. $\phi_1 = 0$, $\theta_1 = 2.064742$.

(i) Is the variance $\gamma_x(0)$ still the same as it was for the models first plotted in (c)?

(j) In one sentence, interpret what you see in (g). How does increasing ϕ_1 or θ_1 affect how the the forecast error decays as k increases?

Estimation

Let's look back at the chicken data one more time. We're going to consider $\mathbf{ARMA}(p, q)$ time series models for the residuals from a linear regression of the chicken prices on an intercept and time. I've computed them for you here and named them `r`, to make sure everyone starts out on the same page.

```
library(astsa)
data(chicken)
r <- lm(chicken~time(chicken))$res
```

2. First, we're going to consider setting $q = 0$ and fitting an $\mathbf{AR}(p)$ model to `r`.

(a) Using the `acf` function, plot the sample partial autocorrelations. Based on the sample partial autocorrelation function, what would you select for p? Accompany your choice with at most one sentence of reasoning.

(b) Setting the order based on (a), fit the $\mathbf{AR}(p)$ model to `r` using the Yule-Walker equations. I do not want you to use the `ar.yw` function, but you can base your code on the the `solve.direct` function. Give the $\mathbf{AR}(p)$ parameter estimates.

3. Now, we're going to consider setting $p = 0$ and fitting an $\mathbf{MA}(q)$ model to `r`.

(a) Using the `acf` function, plot the sample autocorrelations. Based on the sample autocorrelations, what would you select for q? Accompany your choice with at most one sentence of reasoning.

(b) Fit an $\mathbf{MA}(1)$ model to `r` by computing the innovation coefficients $\boldsymbol{d}_n$ using the function given in class, and treating the innovation coefficient estimates $\boldsymbol{d}_n$ as estimates of $\psi_1, \dots, \psi_n$. Give the $\mathbf{MA}(1)$ coefficient estimate.

4. Finally, let's to consider fitting an $\mathbf{ARMA}(p, q)$ model to `r`.

(a) Can we use the sample autocorrelations or sample partial autocorrelations to select p or q? Just give a yes or no.

(b) Using the `arima` function, fit **ARMA**(p, q) models with $p = 0, \dots, 3$ and $q = 0, \dots, 3$, excluding the case where $p = q = 0$. Compute AIC, AICc, and BIC according to the equations that were given in class early on by hand.

(c) Plot the residuals from the AIC-minimizing model.

(d) Fit the AIC-minimizing model to `r` using the `arima` function. What algorithm does the `arima` function use to estimate $\phi_1, \dots, \phi_p$, $\theta_1, \dots, \theta_p$ and σ_w^2 by default? (Unconditional maximum likelihood, unconconditional least squares, conditional least squares…)

(e) On a plot with the x-axis ranging from $2001 + 7/12$ to 2020 and y-axis randing from 60 to 130, plot:

- The observed chicken time series.
- The linear model fit of the chicken time series.
- Predicted mean chicken prices from $2016 + 6/12$ to 2020 based on the model you fit in (e).
- Predicted mean chicken prices from $2016 + 6/12$ to 2020 plus or minus one standard error based on the model you fit in (e). You can get the predictions and standard errors by appling the `predict` function to your `arima` object.