**ARIMA model building – Steps**

The goal of this section is to show how one can go through the process of choosing the best model for a time series data set.

Model building process (steps 1-3 of 6 total steps)

1. Construct plots of xt vs. t and the estimated ACF to determine if the time series data is stationary
	1. If it is not stationary in the variance, make the appropriate transformation
	2. If it is not stationary in the mean, examine differences of xt
	3. Use plots of xt vs. t and the estimated ACF to determine if what you did worked.
2. Construct plots of the estimated ACF and PACF of the stationary series (call it xt).
	1. Match patterns in these plots with those of ARMA models.
	2. Determine a few models to investigate further.
3. Find the estimated models using maximum likelihood estimation.

Example: AR(1) with ϕ1 = 0.7, μ = 0, and  = 1 (ar1\_model\_build.R, AR1.0.7.txt)

Step #1

> ar1 <- read.table(file = "AR1.0.7.txt", header = TRUE, sep

 = "")

> x <- ar1$x

> #########################################################

> # Step #1 and #2

> plot(x = x, ylab = expression(x[t]), xlab = "t", type =

 "l", col = "red", lwd = 1, main = "Plot of

 AR1.0.7.txt data", panel.first=grid(col = "gray",

 lty = "dotted"))

> points(x = x, pch = 20, col = "blue")



> par(mfrow = c(1,2))

> acf(x = x, type = "correlation", main = "Estimated ACF

 for AR1.0.7.txt data", xlim = c(1,20), ylim = c(-1, 1))

> pacf(x = x, lag.max = 20, main = "Estimated PACF for

 AR1.0.7.txt data", xlim = c(1,20), ylim = c(-1,1))



While the PACF plot is not needed here, it is often nice to look at both functions in the next step. Notice that these plots start at lag = 1.

From examining the plot of the data over time and the estimated ACF plot, there is no evidence against the stationary assumption.

Step #2

From examining the ACF and PACF plots above, it appears that an AR(1) model would be appropriate for the data. I am also going to examine a MA(1) model just to show what could happen if the “wrong” model is chosen.

Step #3

> #ARIMA(1,0,0)

> mod.fit1 <- arima(x = x, order = c(1, 0, 0), include.mean

 = TRUE)

> mod.fit1

Call:

arima(x = x, order = c(1, 0, 0), include.mean = TRUE)

Coefficients:

 ar1 intercept

 0.6854 -0.4322

s.e. 0.0730 0.3602

sigma^2 estimated as 1.336: log likelihood = -156.68, aic = 319.36

> z <- mod.fit1$coef/sqrt(diag(mod.fit1$var.coef))

> p.value <- 2\*(1-pnorm(q = abs(z), mean = 0, sd = 1))

> data.frame(z, p.value)

 z p.value

ar1 9.392902 0.0000000

intercept -1.200045 0.2301219

> #Could use confint() function too

> #ARIMA(0,0,1)

> mod.fit2 <- arima(x = x, order = c(0, 0, 1), include.mean

 = TRUE)

> mod.fit2

Call:

arima(x = x, order = c(0, 0, 1), include.mean = TRUE)

Coefficients:

 ma1 intercept

 0.5801 -0.4794

s.e. 0.0661 0.1979

sigma^2 estimated as 1.581: log likelihood = -164.99, aic = 335.98

> z <- mod.fit2$coef/sqrt(diag(mod.fit2$var.coef))

> p.value <- 2\*(1-pnorm(q = abs(z), mean = 0, sd = 1))

> data.frame(z, p.value)

 z p.value

ma1 8.770337 0.00000000

intercept -2.422093 0.01543140

Model building process (continued)

1. For each model chosen, investigate diagnostic measures.
* Examine the residuals to determine if they appear to be “white noise”.

Remember that wt ~ independent (0,). The residuals  were introduced during the forecasting notes.

Examine an ACF plot of the residuals to determine if they appear to come from a white noise process.

If they do, then the model assumption about wt ~ independent (0,) is satisfied.

If they do not, then changes need to be made to the model. There is an autocorrelation pattern in the xt that has not been accounted for by the AR or MA terms. The changes that need to be made correspond to the autocorrelations on the ACF plot. In other words, the residual ACF plot can be used just like the ACF of the original data!

Why can this be done? The answer is coming up!

* Examine a PACF plot of the residuals to determine if they appear to “white noise”.

If they do, then the model assumption about wt ~ independent (0,) is satisfied.

If they do not, then changes need to be made to the model. The changes that need to be made correspond to the partial autocorrelations on the PACF plot. In other words, the residual PACF plot can be used just like the PACF of the original data!

Why can this be done?

Example: Examining the residuals

Suppose the residuals for

  (1)

do not correspond to a white noise process. Note that the p′, q′, and d′ correspond to the AR operator order, MA operator order, and differencing order, respectively.

Using the residual ACF and PACF, a new model for the bt can be identified to be

 (2)

Note that the bar over ϕ and θ is just to differentiate between the parameters in (1) and (2).

Solving for bt in (2) produces  and substituting this into (1) produces



Suppose that a series is wrongly identified as ARIMA(0,1,1): .

Also, suppose examining the residuals identify the model ARIMA(0,0,1): .

The “identification” could be done by examining the residual ACF and PACF. In this case, the residual ACF has a spike at 1 and the PACF tails off to 0. Solving for bt and substituting this into the wrongly identified model produces



Thus the new model is ARIMA(0,1,2).

* Examine the standardized residuals for outliers.

Standardized residuals:



where  is the predicted value of xt for time t using the model.

Some individuals may refer to  as  because only the previous t – 1 xt’s are used directly to calculate the predicted value. However, the model parameters are based on all of the observations so I prefer to use .

If the correct model is chosen, these standardized residuals should approximately follow a N(0,1) distribution. Thus, we would expect about 95% of them to fall within ±2 and about 99% to fall within ±3. Standardized residuals falling outside of this range indicate the correct model may not have been chosen.

If needed, special model terms can be introduced into a time series model to account for these outliers. For example, special cases of transfer function models can be used.

Note that some individuals will refer to residuals as “innovations” in time series analysis.

* Plot the standardized residuals (or residuals) vs. time.

This is another method to determine if there is autocorrelation among the standardized residuals.

* Perform the Ljung-Box-Pierce test to determine if there is evidence against the wt independent assumption.

This performs a hypothesis test on a group of residual autocorrelations.

Ho: ρe(1) = ρe(2) = = ρe(H) = 0

Ha: Not all 0

where ρe(h) for h = 1, …, H is the autocorrelation for the et’s.

Many different values of H are chosen to provide more than one hypothesis test. Some textbooks suggest performing a test with H = 20.

The test statistic for the Ljung-Box-Pierce hypothesis test is



Under the assumption that the null hypothesis is true, Q ~ for large n.

If Q > , then the null hypothesis is rejected and other models should be examined.

If Q ≤ , then there is not sufficient evidence against the null hypothesis. There is not enough evidence that the model should be changed. This is what you want to happen if you are satisfied with your model!

The direct function in R to perform the test is Box.test(). By specifying the argument,type = "Ljung-Box", the Ljung-Box-Pierce hypothesis test is performed for a specified number of lags using the lag *=* argument. By default, the Box-Pierce is performed where type = "Box-Pierce"*.* Its test statistic is



Under the assumption that the null hypothesis is true, Q ~  for large n. Box, Jenkins, and Reinsel’s work suggest the Ljung-Box-Pierce test statistic is better approximated by the χ2 distribution than the Box-Pierce test.

Note that thetsdiag()function to be discussed soon will perform the Ljung-Box-Pierce hypothesis test automatically along with calculating other items.

* Examine the normality assumption for the wt by constructing histograms and Q-Q plots of the residuals or standardized residuals.

The Q-Q plot is a plot of the ordered residuals versus ordered quantiles from a normal distribution. If the points fall on a straight line, this suggests the normality assumption is satisfied. If they do not, then the normality assumption may be violated.

The histogram can be plotted with a normal distribution overlay to assess normality. If there is evidence against the normality assumption (for example, histogram has large deviations from the normal distribution overlay), investigate similar transformations as previously done to solve nonstationary variance problems.

* Perform hypothesis tests for ϕi = 0 and θj = 0 parameters.

If the test does not reject these hypotheses, then a different model may be needed.

Examine the various diagnostic measures in an iterative manner until at least one model satisfies all of the model assumptions!

Example: AR(1) with ϕ1 = 0.7, μ = 0, and  = 1 (ar1\_model\_build.R, AR1.0.7.txt)

Step #4

AR(1) model:

> #########################################################

> # Step #4

> tsdiag(object = mod.fit1, gof.lag = 20)

> #Could also get the Ljung-Box-Pierce test this way.

> #Box.test(mod.fit1$residuals, lag = 20, type = "Ljung-

 Box")

> #Box-Pierce test

> #Box.test(mod.fit1$residuals, lag = 20)



> par(mfrow = c(2,2))

> pacf(x = mod.fit1$residuals, lag.max = 20, main =

 "Estimated PACF for residuals using ARIMA(1,0,0) fit",

 xlim = c(1,20), ylim = c(-1,1))

> hist(x = mod.fit1$residuals, main = "Histogram of

 residuals", xlab = "Residuals", freq = FALSE, col = NA)

> curve(expr = dnorm(x, mean = mean(mod.fit1$residuals),

 sd = sd(mod.fit1$residuals)), col = "red", add =

 TRUE)

> qqnorm(y = mod.fit1$residuals, ylab = "Residuals",

 panel.first = grid(col = "gray", lty = "dotted"))

> qqline(y = mod.fit1$residuals, col = "red")

> par(mfrow = c(1,1))



MA(1) model:

> #MA(1)

> tsdiag(object = mod.fit2, gof.lag = 20)



> par(mfrow = c(2,2))

> pacf(x = mod.fit2$residuals, lag.max = 20, main =

 "Estimated PACF for residuals using ARIMA(0,0,1)

 fit", xlim = c(1,20), ylim = c(-1,1))

> hist(x = mod.fit2$residuals, main = "Histogram of

 residuals", xlab = "Residuals", freq = FALSE, col = NA)

> curve(expr = dnorm(x, mean = mean(mod.fit2$residuals),

 sd = sd(mod.fit2$residuals)), col = "red", add = TRUE)

> qqnorm(y = mod.fit2$residuals, ylab = "Residuals",

 panel.first = grid(col = "gray", lty = "dotted"))

> qqline(y = mod.fit2$residuals, col = "red")

> par(mfrow = c(1,1))



Notes:

* Everything appears to be fine with the AR(1) model. There is a significant (barely) partial autocorrelation at lag 5. There are some people who will fit a model of the form:

xt = ϕ1xt-1 + ϕ5xt-5 + wt.

Notice that ϕ2, ϕ3, ϕ4 are assumed to be 0 here. I prefer not to do this unless there is a known “seasonality” to the data. We will discuss seasonal ARIMA (SARIMA) models later in the course.

* Similar to the previous note, there is a barely significant autocorrelation at lag 5. Some people may add a θ5wt-5 to the model, but I recommend against it unless there is a seasonality aspect to it.
* Problems with the MA(1) model were detected. There are some autocorrelations and partial autocorrelations found to be not 0. The Ljung-Box-Pierce hypothesis test also detects some autocorrelations are not 0.
* On p. 4 in the notes, I performed a test of Ho:ϕ1=0 vs. Ha:ϕ1≠0 and the test rejects Ho. The test statistic was



* On p. 5 in the notes, I performed a test of Ho:θ1=0 vs. Ha:θ1≠0 and the test rejects Ho. The test statistic was



* What if you do not reject?
* The number of lags displayed on the plot of the Ljung-Box-Pierce test p-values can be changed using the gof.lag argument intsdiag(). Unfortunately, there is not an argument to control the lags displayed for the ACF plot produced by tsdiag().

Model building process (continued)

1. Using each model that gets through 4), pick the best model based upon model parsimony and the AIC. The best model corresponds to the one with the smallest number of AR and MA parameters and the smallest AIC.
* Principle of parsimony – choose the model with the smallest number of parameters
* Akaike’s information criterion (AIC)

The AIC is

-2 + 2(# of parameters estimating)

where  is the log-likelihood evaluated with parameter estimates and  is a vector of the estimated w’s (residuals).

Because the likelihood function is a measurement of “how plausible” a set of parameter estimates are, a criteria of model fit is to have  be as large as possible. Multiplying by -2 means we want
-2 to be as small as possible.

The 2×(# of parameters estimating) part serves as a “penalty” for the number of parameters in the model. This is needed because generally  can be made larger by adding more parameters (more AR or MA terms).

In summary, the model with the smallest AIC is the “best” model to use (according to this criteria).

Because models are fit to the same data set, there are some parts of the likelihood function that will be the same for each model. This will cause some software packages (and packages within R) to drop these parts of the likelihood function in the calculation of the AIC. Thus, different software packages can result in different AIC values so please be careful in comparing them across software packages!

* Forecast error – choose the model that does the best with forecasting

Choose the model that has the smallest forecast error. How can you do this if the future values are not known?

Estimate the model without the last c observations where c < n. Forecast the c observations and find the mean square error. The smallest MSE corresponds to the best model using this criterion.

Problem: You are no longer using all of your observations in the model.

Example: AR(1) with ϕ1 = 0.7, μ = 0, and  = 1 (ar1\_model\_build.R, AR1.0.7.txt)

Step #5

Because the MA(1) model did not “pass” the various diagnostic tests, the model should not be considered further. For illustrative purposes, I am considering it further here.

Both models have the same number of AR and MA parameters so parsimony cannot be used to choose between the models.

The AIC was shown previously in the R output for the arima() function. Below are the AIC values again by extracting parts of the model fit object.

> mod.fit1$aic

[1] 319.363

> -2\*mod.fit1$loglik+2\*3

[1] 319.363

> mod.fit2$aic

[1] 335.9806

According to this criterion, the AR(1) model is the best because it has the lowest value.

Model building process (continued)

1. Begin forecasting!

This is not really a “model building” step, but I include it here because this is a main purpose of creating the model.