Time Series Modelling and Forecasting with Applications in R

Prof. Sudeep Bapat

Shailesh J. Mehta School of Management

Indian Institute of Technology Bombay

Department of Applied Statistics/Finance

Week 11

Lecture 55: Practical Session in R - 11

Hello all, welcome to this course on Time Series Modeling and Forecasting using R, ok. So, we are almost done with Week 11 here, and then after Week 11, there will be one last week left, and this would be a practical session within Week 11. Now, again, just to give you, as always, my habit is to start with a very quick refresher, right? So that people do not seem disconnected with what has been done in the theory. So, throughout this week, we studied the idea of non-linear time series processes.

So, we started with different kinds of regimes, then specific kinds of models, let us say TAR model or STAR model, SETAR model, right? And in the last session, we talked about the third kind of regime, which is called Markov switching models, right? So, I think now is the time to connect all the theory and then get our hands dirty using some, let us say, simulated data or real data. So, we will try to implement all the models we have seen this week on some set of simulated data or rather real data across multiple areas, etc., ok. So, again, get your laptops open, get your R worksheets running, and by the way, this worksheet requires all these packages.

So, since non-linearity is not very easy to understand, the first thing we do is to install all these packages. So, let us say TSA, forecast, quantmod, TSDYN, stats, deSolve, tseries, chaos, etc. Ok. So, again, the very first step would be to go into Tools and click on Install Packages, then try to install each and every package here, ok.

And again of course, once you have installed all the packages do not forget to call them in the R environment also using a library command ok. So, let me quickly do this for each and every So, TSA, forecast, quant mod, TS, DYN, stats, desolve, T series chaos etcetera ok. And now once you have made sure that all the packages are there which are

required then you can start running the codes ok. So, now the first thing is as always set dot seed 1, 2, 3 again if you want to regenerate

then r would start from the same point, but ensure some randomness. And here again, as always, you can give any random number that you want. So, we will give 1, 2, 3 for the time being. So, set dot seed and then 1, 2, 3 here. So, now the very first part of this code would be running some examples on simulated data.

So, part 1 would be focusing on simulated data example and simulating which model exactly? So, simulating a self-exciting threshold AR process or rather SETAR model ok. So, self-exciting TAR model or rather self-exciting threshold AR process or in short SETAR process. So, the very first exercise we will do is depending upon some fixed parameter values we will try to simulate from a SETAR process.

So, for that we will fix the sample size first. So, let us say n is 500 and then all the parameters we will fix. So, let us say threshold value. So, it will specify some threshold value. So, being a threshold kind of a model, you would always specify some threshold value.

So, let us say we will keep the threshold value to be 0 for the time being. and then we can define all the parameters. So, phi 1 is the AR coefficient for regime 1, phi 2 is the AR coefficient for regime 2. So, my phi 1 happens to be 0.8, my phi 2 happens to be minus 0.5 and sigma is nothing, but the standard deviation of the error term or the noise term which is assumed to be 1. So, once you fix all the parameters then you can sort of go ahead and simulate from that ACETAR model ok.

Now, beforehand I will do one small exercise is that y and then arrow numeric n. So, I am considering that the value that my y is taking is nothing, but the value that my n is taking initially and now. So, this is a small loop kind of a structure where I am generating the SETAR process. Now, again remember one thing that in SETAR you have two regimes right. So, depending on one threshold value. So, given one threshold value it splits into two regimes.

And within each regime, I am trying to fit some AR model. So, again, just to very quickly explain what is going on in this chunk of the loop that you see here, let us say t goes from 2 to n. So, t is a pointer or rather a counter. So, t goes from 2 to n, where n is the last value of the time series. So, if y_t minus 1 is less than or equal to the threshold, it will be in the first regime, right? Else, it will be the second regime.

And this is exactly as per the SETAR definition. So, again, if you vaguely remember the model structure, we have seen in the last couple of sessions about the SETAR process, then this is exactly what happens there, right? So, you look at the threshold value, and again, the threshold value here is nothing but the past value of the time series itself. So, y_t minus 1. So, if that threshold value is less than or equal to the fixed threshold value, which is 0 in this case, then the time series will be in regime 1. Otherwise, if y_t minus 1 exceeds the threshold or exceeds 0, then you will switch to regime 2.

Makes sense? And within each regime, what model do you have? So, within each regime, I have an AR structure or a simple AR structure. Makes sense? So, let me run this loop.

So, again, generating the SETAR process should not be difficult, and again, understanding what goes inside the code or what goes inside the loop should not be difficult now, ok. So, let me run this overall for loop. And now, let me show you the plot. So, the first plot would be the plot of the simulated SETAR process, which looks something like that. So, let me zoom in.

So, this is exactly how the simulated SETAR process looks. So, again, this process is not a whole lot different from any of the other simulated processes we have seen, let us say, in the last practical session or the one before that, etcetera. So, one cannot basically, or rather, differentiate a simulated SETAR process from, let us say, any of the other AR processes, right. At the end of the day, it is a simulated process. So, one key characteristic could be that there could be some hidden non-linearities in the underlying data, ok.

Alright, so this would be a simulated SETAR process, and then I will include an AB line on the threshold value just to sort of specify or rather focus on the threshold variable, which is 0 here. So, the horizontal red line that you see is in fact on 0, which tells you the threshold value in the underlying SETAR process. So, let me close the graph, and now, the second kind of model we will try to simulate is a TAR model. So, the SETAR model is done, and then the second model we will try to fit would be a TAR model or a threshold autoregressive model. to the simulated data, ok.

So, again what we will do is we will again pick up the same vector y right and then give it a slightly different name let us say tar underscore model underscore sim and then you have this inbuilt function in R and one of the packages we loaded earlier is called as tar. And this is nothing but the syntax part. So, what goes inside the tar function is nothing but you specify the time series value which is y and then alongside you specify p1 equals 1, p2 equals 1, d equals 1 and a is the threshold. So, now let me write this and then I will

show you a quick summary of the fitted tar model. So, this is the summary of the tar model and this is rather the

fixed all the fixed parameter values we have taken. So, P1, Q1, D, QR1, QR2 etcetera and the overall length ok. Alright and then the key part is once you fit any of the modules you would always check for the residuals right. So, I can do that by plotting the residual diagnostics and then this check residuals command in R is kind of very helpful to do that. Now, we will be checking the diagnostics for which model I mean the TAR model.

So, check residuals applied on the tar underscore model underscore sim ok. So, once you run this then it will give you three kinds of different plots let me again zoom in and these are the key assumptions that we are making right. So, if you are kind of into time series modeling let us say beat arema or beat tar beat star beat se tar whichever model, but again once you are through with the model building you would always ensure that the residuals follow normality. So, which would be given by the histogram.

So, here you can clearly see that the residuals are indeed following a symmetric bell-shaped curve, which might resemble a normal distribution. Now, of course, on top of that, you can do all sorts of hypothesis tests—let us say Shapiro-Wilk or Jarque-Bera, some sort of formal test to confirm the normality. But again, if a histogram looks as solid as that, then probably you need not even go for some of the hypothesis tests, okay? On the other hand, this is nothing but the distribution or a raw plot of the residuals, and here you can see that the residuals do not have any inherent pattern in them. So, they are completely random, and variance is not an issue here.

So, variance would be assumed to be constant, right? Which can also be seen using the ACF plot. So, in the ACF plot, you can see that almost all the spikes are within the bounds, which is a clear indication that all the assumptions about the residuals are met. Make sense? So, here again, one small point is that if you see any discrepancy or some disconnect in any of the plots—let us say normality, changing variance, or autocorrelation—then again, you have to go back, push back a step, and then try to, let us say, propose a slightly different model. Then fit that model and check for the residuals. So, you have to do some back-and-forth kind of exercise.

But once you are assured that all the assumptions are being met—which they are in this case—then you can go ahead with the forecasting. So, I will close the residual plots, and now the last thing is forecasting with the TAR model. So, we will give it a name—let us say, sim_forecast—and then again, predict is an inbuilt function, and we have used this

predict function multiple times before, predicting which model. So, we want to predict the TAR model and how many time steps ahead. So, let us say 20 time steps ahead.

So, let me run the forecast and give you the plot. So, plot and then along with the lines. So, let me close this first. Let me recreate the plot. And then I will click this as yes.

By the way, I will show you one trick. So, as of now, within the plotting window, you must have seen that R is giving you multiple plots in the same window. So, how do you correct that? So, let me scroll down. I think I have the code here somewhere for that.

Or rather, if not—or rather, if not—yeah, for example, here. So, you have to run this line. So, that you see in line 101. So, this would again go back to having one plot in a single window, right? Because again, let us say that if you do not want multiple plots in the same window, then why would you want smaller plots, right? OK.

So, again, I do not want to have all the individual plots in a small plot. So, again, you might want to run this. So, this tells R that I only want one plot per row. OK. So, PAR and then MF rho equals concatenate 1 comma 1.

So, let me run this, OK? And once you run this and if R is not throwing any error, then you are good to go. So, again, scroll up to the forecast plot; this is exactly what we were doing. So, again, let me rerun just in case the forecast code and then the plot, OK? So, now I think this is much better. So, this is the actual plot of the simulated data, and this is the

So, let me zoom in just to sort of tell you exactly how the forecasts are looking. So, again, this is the simulated data up to this point, up to 500, and the blue dots that you see are nothing but the forecasted values using the TAR model, or rather, the TAR model forecasts. So, again, let me go back to the code. Now, once we are through with the simulated data, we will get our hands dirty with exploring some real data. So, part 2 in this code would be a real data example, taking the example of the Nile river data.

So, for that would you load the Nile data set first. So, let us say you load the Nile data set first using data and then you specify the Nile as a name and again just to quickly show you as to what this Nile data set contains. So, again I can run question mark on Nile and it should give you a small description on the right hand side. This Nile dataset gives you the flow of the river Nile. So, measurements of the annual flow of the river Nile at a puddler location called as Aswan and over all these years.

So, 1871 to 1970 in let us say these units with apparent change point near 1898. So, this is such a data the Nile data that if you talk about the flow of the river Nile it has a very strong change point in the year 1898. Okay, and then we will, so I will show you the plot, do not worry, but then the change point kind of points to a fact that probably a linear kind of a time series structure may not be good here, right, because if the time series or the behavior is changing, you have a strong change point somewhere in between, then probably this indicates that you have a threshold value and then you have some regimes, okay, and the time series is transitioning between these regimes, okay. So, again going back to the data which is Nile and then let us say I will give it a name to extract the only the flow data. So, I will give it a name as Nile underscore flow which is nothing but as dot numeric applied on Nile and then exploring the data.

So, let us say summary variables, some preliminary summary variables of the Nile flow data. So, these are the values. So, let us say the mean flow is 919 of course in those units that you see in the right hand side. This is the minimum flow. This is the maximum flow.

So, 1370. How much is Q1? How much is Q3? How much is Q2? Etc.

And here I will just quickly show you the histogram of the underlying data. So, the Nile flow data looks like that. So, let me zoom in one more time. So, here clearly you can see that the majority of the flow values lie between, let us say, 600 to 1200, or rather 1300. And then these two could be outliers here.

So, one on the lower end and then the other one on the upper end. So, this is a broad histogram of the flow data, which is the Nile flow data. So, now once we have seen, once we are through with understanding a few ideas from the histogram, we can start putting forward some models for the data. So, the very first model we will try is checking for the scatter against linearity. Now, here what we will do is

Let us say if you assume a linear model and then you are kind of assuming that linearity is not good enough, then you want to put forward some non-linear model. Then how do you check for a SETAR model against a linear model using some hypothetical test? So, this SETAR test, which is an inbuilt function in R, exactly does that. So, how exactly? So, I will show you. So, the SETAR test applied on the Nile flow data, m equals 3, n boot equals 400. This is part of the syntax; let me run this.

And then it will probably take a short while to run because the SETAR test sort of takes into account all the iterations in the background and then throws you the best possible

models. So please give it half a minute to run. Don't panic. So in some situations, if you see a stop sign here, then now it's done, of course. But if you see a stop sign here, then you see that a program is running.

So, if a program is running, you should not do any other codes there. And then you should stop once the stop sign is gone, basically. Alright, anyway, so this is a test of linearity against two models. So, ACETAR-2 and then ACETAR-3. And how do you read this? The first row gives you 1 versus 2.

So, linearity against S e to the power 2 and then second it gives you linearity against S e to the power 3 and it gives you the corresponding p values. Of course, it gives you a test statistic values, but we are not concerned about the test statistic values, but more about the p values. So, in the first two of you see the p value is slightly significant of course, not at the 0.05 level, but let us say the 0.1 level. So, if you fix alpha to be 0.1 this could be significant of course, this is not significant. So, what it means is that a CETAR 2 model could be slightly preferred.

as compared to linearity, but of course, Acetar 3 model might be overfitting. So, Acetar 3 model is not preferred because if you see here p value is not less than 0.05 or 0.1 right, but since this p value is less than 0.1, we can sort of conclude that Acetar 2 model might be preferred as compared to a linear structure. So, hopefully the idea of the ACETAR test is clear that ACETAR test again just to summarize gives you a sort of a comparison between linearity and any of the non-linear models. So, in this case ACETAR model. Now, how do you select the ACETAR model?

So, once you make sure that there could be a possibility of putting forward some sort of an ACETAR model on the 9 flow data, then how do you select the orders? So, I have run this line 61 again probably just wait for less than half a minute and then this throws you this table and probably the first attention point of view from in your situation should be the last column. So, last column gives the pooled AIC value and basically you have to scroll down the line and then basically find out the model which has the least AIC of course, ok. And out of all these 10 different iterations probably the very first model corresponds to the least AIC which is 1, 2, 3, 8.051.

And then you sort of trace back in that the first row and then these are the parameter values. So, the threshold value is 912 by the way. So, corresponding to the first model, the threshold value is 912, the other parameter, the delay parameter is 2, etc. But once

you are sure that which level or rather which row to focus on, then again you can create the model now. So, how do you create the SETAR model on top of the Nile flow data?

You can use again the same function called SETAR, but apply it on the Nile flow data. But here you fix some of the parameters which you get from this table by the way. So, how much is the delay? So, delay is 2. How much is the threshold?

So, threshold is 912 etcetera. So, once you run this, then you can run the summary of the model and then this is exactly the model that you have along with all the estimated parameters. So, phi L1, phi L2, phi L3. By the way, here you can clearly see that you have two regimes. So, one is a low regime.

And this is usually the syntax that R follows. So, L stands for low regime, and H stands for high regime. And within each regime, I have an AR structure containing three coefficients. So, phi 1, phi 2, phi 3 in L, and similarly phi 1, phi 2, phi 3 in H. Does that make sense? So, this is exactly what R throws at you.

So, estimated coefficients—this is the threshold variable which takes that form. The threshold value is 912, etc., right? So, now at this point, we successfully try to model the Nile flow data using a set R kind of model and then the forecast, right? So, one can always forecast it down the line and then use the predict function again for how many time points down the line. So, let us say 10 time points down the line.

So, N dot ahead is 10, and then if you want to output the values, I can always write down Nile underscore forecast. So, all these are the forecasted values, or rather the forecasted flow values down the line. Does that make sense so far? So, now another data set. So, let us say enough about the Nile data set.

So, let us say there is one more very famous data set which people analyze is called a sunspots data. So, firstly what exactly are sunspots? So, sunspots are nothing but small black spots on the surface of the sun and then these are sort of capable or rather very helpful in understanding the different wave mechanism. So, let us say electromagnetic waves right. all these kind of ideas.

So, sunspots data is, so a lot of physicians kind of use this data to analyze multiple things. So, again here we make use of the sunspots data. So, I will show you the plot of the data first. So, plot dot ts of sunspot dot year and this is exactly how the sunspots data looks like. Let me zoom in.

And then here you can clearly see that there might be a very very slight trend, slightly upward trend, but again you can see some seasonality or rather some cyclicality here. So, again these are some of the features that one should always keep in mind when you see any of the time series plot. All right. So, now test for linearity. So, again we have not discussed these in the theory, but I thought that I will just throw these two tests in the practical session.

So, these are some hypothetical tests. Again, do not go into detail as to what the test statistic is, etc. But the names are the Keenan test and then PSAY, so the psi test. So, the Keenan test and then the psi test applied on sunspot here.

And then what these two tests would give you is that if you run these two tests, then of course, they will give you a p-value. So, again for the Keenan test, the p-value is pretty small. Again for the psi test, the p-value is pretty small, 6.6 into 10 to the power minus 12, which means that you can actually reject the null hypothesis of linearity. So, since you are testing for linearity, my H naught or my null hypothesis is that the model is linear, which you are sort of rejecting here because the p-value is very, very small. Make sense? So, in fact, both these tests are pointing to the same idea or same notion that the underlying time series model need not be linear, and it is a good idea to explore some non-linear structures, OK?

Now, what exactly is this? So, we will try to fit some model here, right? So, we will try to fit some model using this AICM technique. So, again, I am pretty sure that we have not discussed this in the theory, but this AICM technique, what it does is it sort of Again, you can see a for loop going from 1 to 9.

So, it takes each and every iteration and then gives you the best possible model. So, I will run this for loop and then column names, row names and then AICM table. So, this is exactly what the AICM table looks like. Again, kind of a similar looking table as we saw earlier, the column to note here is the AIC column and then one should identify the minimum AIC as much as possible. And here if you scroll down, I think the minimum AIC is nothing but given by the 9th row, right, which is 2221.

So, in between we had a slightly lower AIC here which is 2226, but 2221 is of course, the least AIC. So, one should go with that. And once you identify the model, then you have to read the ninth row to identify the other orders. So, what is R, what is P1, what is P2, etc. And then lastly, we will be fitting the

model containing all these values on the sunspot data ok. So, P is 9, D is 9, A is 0.15, B is 0.85 etcetera and then this is done and then now once you fix the parameter values now we can fit some in some sort of a tar model on that data ok. So, sunspot dot tar dot best would be fitting a tar model on the sunspot data and all these parameter values are from the table that we saw just now. So, let me run this then let me run the plot now again as you see that you see some of some error in plot dot nu that figure margins are too large right and probably you have covered this some of the earlier practical session as well that if r throws an error like this you have to basically run this line. So, par mar equals 1 1 1 1.

So, if you run this And then again go back and try running the tsdiag command. Now again I think it's doing the same error but again. So, idea is that it should throw you. So, I think this line should sort of get rid of the error which is not happening here.

That is ok. But ideally that should happen. So, this TS diag should output some plots, some visual plots. Anyways, let us go for prediction, forecasting, etc. So, let us say you can forecast down the lines using the sunspot data.

And then how many time points are the lines? So, let us say 10. And then if you want to extract the predicted values, so sunspot dot pred, then you can have a plot. Again, it is saying figure margins are too large. Let me try to run this one more time.

And let us see if this happens now. So now we have the base plot, and then we have the added line. So, this is the sunspot data. This is the forecasting window. This is the upper limit.

This is the lower limit. Now, let me zoom in again here. If you see that R is again giving one small plot rather than one single plot in one window. So, how do you counter that? So, I can again run this line and let me go back to creating all these plots one by one. So, plot the predicted values, then the lines, then the actual forecast along with the interval.

Now, let me zoom in, and this would give you a better picture. So, this is the actual sunspots data along with the TAR forecasts. So, the dotted, or rather the dashed line, gives you the TAR forecasts. And again, probably I will skip this last bit because the time is not there. But again, as a small disclaimer, given the last practical session as well, a strong suggestion would be to go ahead and then try to complete this entire worksheet from your end.

And then by the way, this one piece which is sort of remaining is fitting some linear AR model and then how do you identify the order of the AR model. So, all those things. But

since this is not very much related to let us say non-linearity, so probably it can also be best if you skip it basically. So, think that the code has ended here up to line 101 because we have kind of fitted two different models. So, TAR model and then SETAR model.

And then we have seen that we have fitted on two different data sets, so the Nile flow, sunspots, we have forecasted down the line, we have compared, etc. So, I think this would be a strong understanding about the entire idea about non-linearity. And now we are entering into week 12 from of course the next session and then the idea there would be to explore the ideas about machine learning integration or rather let us say deep learning integration with time series. So, application of time series connecting with machine learning or deep learning, reinforcement learning, neural networks. So, again I am pretty sure that in the very last week I have kept predominantly just as a closing week which would be really interesting because it sort of connects through all the recent ideas about let us say ML or AI or data science or integration of all these three and application from a time series point of view.

Thank you.