Time Series Modelling and Forecasting with Applications in R

Prof. Sudeep Bapat

Shailesh J. Mehta School of Management

Indian Institute of Technology Bombay

Week 06

Lecture 27: ARFIMA Processes

Hello all, welcome to this course on time series modeling and forecasting using R. Now again, as all of you probably know, this week we are focusing more on a slightly different kind of time series model called RFEMA processes. The full form of RFEMA is auto-regressive fractionally integrated moving average processes. Now again, if you vaguely remember—and this is a very quick review—in the last lecture, we talked about a particular kind of time series characteristic called persistence in time series. And again, there we talked about several examples where persistence could be observed.

Now again, what exactly do we mean by persistence? So the idea of persistence in any time series is nothing but its long-term behavior—when the time series retains memory for quite some time before it reverts back to the mean. And if a process is not persistent, we'll call such processes anti-persistence time series, right? And again, in the last lecture, if you remember, we took up many examples from different areas, such as stock prices or environmental sciences. And then we talked about a particular example in each category, such as global warming.

So in all these examples, one can actually witness some persistence, right? Now again, just to quickly revise the idea of visually checking persistence in a time series, let's say you have a graph of a stock price and the movement of that stock price on a daily basis. And let us say initially the stock price behaves like that, and due to some random shock—which could happen because of some positive or negative news about the stock—the stock behaves in a very erratic manner, and one can actually see a lot of large fluctuations like this due to that random shock event. Now, if the time series is supposed to be persistent, then it retains the memory for quite some time.

So a similar kind of a structure could be seen even after a few lags and then eventually it should die down very very slowly. As opposed to that, if you have a time series which is not persistent, then what should happen? Again, let us say the stock is behaving like that and you do some random shock, it picks up and then you can see a lot of fluctuations. But since the time series is not persistent or in other terms anti-persistent, it will not take much time to sort of revert back to its mean. So again, you can see the same tendency after a short period of time.

So, the above example is of a persistence time series while the below example is for a anti-persistence time series. And one last thing here just to remember is that all the models we have seen so far, so be it let us say ARMA or ARIMA or SARIMA, they sort of deal with memories which are shorter terms, right? or time series processes which are not persistent, right? And why exactly because due to the structure of the underlying model let us say ARMA or ARIMA whatever the ACF or the auto correlation function sort of dies down very nicely in an exponential manner rather than slowly dying down. So this is a small difference between visually also how do you check if the ACF is sort of behaving or indicating a persistent nature in the time series or an anti-persistent nature. And hence there should be a need of a slightly different set of processes which could sort of model the long term memory of the time series or the persistence nature of the time series.

And hence in today's lecture we will talk about specifically this model which is Arfima and then we will bring in some theoretical angles and then we will try to build the Arfima process and then when exactly an Arfima process becomes different from let us say an Arima process or an Arma process etc. So all these things we will discuss in today's lecture. Now, again just to reiterate one last time as to what the full form of ARFIMA is. So, autoregressive fractionally integrated moving average. Now, this term fractionally integrated is sort of different from ARIMA.

So, ARIMA stands for autoregressive integrated moving average, and ARFIMA stands for autoregressive fractionally integrated moving average. Now, firstly, we will soon see what you mean by this fractionally integrated as opposed to simply integrated. But again, just to repeat one last time, the motivation behind developing such a process, which is called ARFIMA, is something like this. So, ARMA models are only able to capture short-term dependence. We discussed this in the last lecture also.

The dependence between observations decreases rapidly as the time lag increases, right? So, if you want to visualize a typical ACF of an ARMA process—now, let us say these are the hypothetical bands—then initially, even if the correlations are significant, they should eventually die down, and then all the correlations would be not significant after a very short period of time. So, such a tendency is characteristic of an ARMA process, etc. But exactly opposite to it, if you see significant correlations outside the bands over a longer period of time, then you require, let us say, probably an ARFIMA process or something like that, OK? And similarly, on the other hand, if you talk about ARIMA processes.

So, integrated ARMA models reduce to a stationary process after a short-range dependence after a finite number of differences, right? So, if you have an ARMA process, then initially, what we do is we difference the process to make it stationary, right? And then, the order of differencing in terms of an ARMA process is an integer, right? So, the order of differencing, which is nothing but the d parameter—if you remember—has to be an integer. So, let us say 1, 2, 3, or something like that.

So, these are the few characteristics of the already discussed processes. So, let us say ARMA or ARIMA, etc. Now, particularly if you talk about where exactly the ARMA models fail, right? So, ARMA models hence fail to capture any long-range dependence as their ACF decays exponentially, right? So, if the correlations are decaying exponentially, then how do you suppose to capture any long-term dependencies or long-term persistence in a time series, right?

So, this would be the question, right? On the other hand, if you talk about an ACF from an ARFIMA process, that decays pretty slowly and it sort of preserves the long-term memory in the underlying time series. Hence, ARFIMA processes provide a much-improved fit and better predictions in comparison to ARMA processes for time series possessing long memory. So, the whole idea here is, I guess, talking about if you want to capture any long memory or, let's say, persistence in the time series, then models like ARMA or ARIMA. So, ARIMA is nothing but integrated ARMA.

So, they sort of fail because the ACF decays exponentially and hence they can't retain the long-range dependence property. And as opposed to that, if you talk about a fractionally integrated process, such as ARFIMA and probably its extensions, then such models prove to be better in sort of capturing that long memory in the underlying time series process. So, initially, just to start things off, we will talk about a particular ARFIMA process which is called fractionally integrated noise. Now, again, this could be a very starting point sort of thing because if you look at the orders here, the orders are nothing but 0, d, 0. So, fractionally integrated noise is nothing but an ARFIMA process with orders 0, d, 0.

Now again, as before, just a couple of things to mention here: these orders, 0 and 0, point to the AR structure and the MA structure correspondingly, and then this middle structure or middle parameter, which is D or the middle order D, is similar to any ARIMA process. So, this tells you how many times one should difference the process. But here, we have a subtle difference. So, we will soon see that the values we can take can also be decimals or fractions. So, the values we can take need not be only integers, as opposed to ARIMA.

Alright, so we are trying to slowly build this ARFIMA process. And then again, I would strongly suggest that if you are not comfortable with any of the notations or any of the procedures in this particular lecture, then probably review the video, let us say, multiple times or read the slides multiple times to get the idea of how ARFIMA and the orders of ARFIMA are different from, let us say, the orders of ARIMA. So, let us say yt is a fractionally integrated noise or, in other terms, a fractionally integrated ARMA process of, let us say, order 0, d, 0, and the essential part here is that d should lie between, let us say, minus half and half. Okay. So, we will see first, as a first case, what would happen if d lies between minus 0.5 and 0.5, okay.

$ARFIMA(0, d, 0)$ process.

Y_t is a fractionally integrated noise or fractionally integrated ARMA process of order (0, d, 0), with $-0.5 < d < 0.5$ if,

y_t is a stationary solution of $(1 - B)^d y_t = e_t$, where e_t is a usual White Noise process with variance σ_e^2.

If yt is a stationary solution of something like this. So, 1 minus B to the power d applied on yt equals et, and what is et? So, et is a usual white noise process with variance, let us say, sigma squared e, which is fixed, okay. Now, before we talk deeply into any of the intricacies and any of the notations, I will quickly remind you what you mean by B firstly, right? So, B is nothing but the backshift operator, right?

$$(1 - B)^d y_t = e_t \Rightarrow y_t = (1 - B)^{-d} e_t$$

The difference operator $(1 - B)^{-d}$ is defined as,

$$(1 - B)^{-d} = \sum_{j=0}^{\infty} \psi_j B^j$$

Where, $\psi_j = \frac{\Gamma(j+d)}{\Gamma(j+1)\Gamma(d)}$.

So, if you apply B on yt, this should actually give me something like yt minus 1. Similarly, what would happen if you apply 1 minus B on yt? So, 1 minus B on yt is nothing but yt minus yt minus 1. So, essentially 1 minus B is nothing but nabla, and nabla is nothing but the differencing operator. So, can you see here?

So, this 1 minus B essentially is nothing but the nabla operator, right? Similarly, as we saw in the ARIMA lecture, one can actually continue differencing, right? So, what would happen if you take 1 minus B squared applied on yt? This would be nothing but 1 minus B applied on 1 minus B applied on yt twice, right? So, this would be nothing but 1 minus B applied on yt minus yt minus 1. So, what would this give you? So, this would be nothing but yt minus yt minus 1 minus, let us say, yt minus 1 plus yt minus 2.

So, here what we are doing again—just if you note carefully—is that if you open the brackets, then the first set would be multiplied by 1. So, it will stay the same. So, yt minus yt minus 1, and then minus what would happen if you operate B on this collection. So, if you are operating B on any time series process, then essentially you have to reduce the subscript by 1. So, if you apply B on this, it will be yt minus 1, but then minus minus becomes plus, and then yt minus 2.

So, eventually, I can combine some notations here. So, this would be y t minus 2 times y t minus 1 plus y t minus 2. And essentially, this thing is nothing but you are basically differencing the y t process twice. So, similarly, as we saw in the ARIMA course or ARIMA module, one can actually keep on differencing the underlying time series process. So, let us say 1 minus b to the power 3 or 1 minus b to the power 4, or in general, if you take a notation of d.

So, let us say 1 minus b to the power d operated on y t means that you are sort of differencing y t d times. Now, coming back to the RFE mass structure. So, what is different here? So, the only thing which is different is that the order of differencing can also take a fraction. So, here specifically, as a first case, we are considering that this D, what would happen if it lies between, let us say, minus half and half.

So, probably what would happen if this lies between minus half and half. So, 1 minus b to the power d operated on y t has to be equal to some white noise term, as seen before in the last slide. So, this implies I can take this term on the right-hand side. If I take this term on the right-hand side, what would we get? So, y t would be equal to 1 minus b to the power minus d and then applied on e t. Now, here, the whole idea behind the ARFIMA process is the convergence of this operator.

So, how fast does it converge or how slowly does it converge? So, you might want to revise very basic convergence ideas, right? So, when would this structure converge very fast? So, for what values of d? Or, in other words, for what values of d would it not converge that fast?

Would it converge slowly, right? So, this is the whole idea, right? Because, if you remember, the whole idea here is to retain that long-term persistence in the time series, right? So, for what values of D would it have a long-lasting impact, and for what values of D would it not have a long-lasting impact? So, this is the whole idea behind designing the ARFIMA process and the implications of that.

Now, let us say you define this difference operator in slightly different terms. So, let us say the difference operator 1 minus B to the power minus d could be defined as something like this, right? So, you have an infinite sum of psi_j B to the power j. Now, B to the power j all of you know, but what is psi_j? So, psi_j is given by this term, right? And, by the way, this term comes from a binomial expansion because if you want to expand this, right?

So, one can actually expand this using a binomial expansion. So, something like 1 minus b to the power of something, right? And in this case, it is minus d. So, if you sincerely take a pen and paper and then sort of apply a binomial expansion, you should essentially get this, where the coefficients are nothing but this. And what exactly is this notation? This is nothing but gamma notation.

So, probably many of you might know that gamma of alpha is nothing but alpha minus 1 factorial. So, in other words, this coefficient can be translated into a set of combinations that you see in a binomial expansion. So, n choose r or n choose r plus 1, whatever. But any combination structure can be written down in terms of factorials and then it can be sort of written down in the gamma notation. So, gamma of alpha is nothing but alpha minus 1 factorial.

So, what would be gamma of 3? So, gamma of 3 would be nothing but 2 factorial, which is nothing but 2, right? Or gamma of 4 would be 3 factorial, which is nothing but 6, etc., all right? So, essentially, I can write down this infinite expansion because this would be an infinite expansion, right? If you apply the binomial expansion, this would be exactly equal to an infinite sum of some coefficients and b to the power j, where these coefficients happen to be nothing but this term here.

Now, further, what would happen if d is strictly between minus 0.5 and 0.5? Then, in that case, a stationary solution of the previous equation becomes something like this. So, while t happens to be an infinite sum of the same coefficients, but then e t minus g. Now, e t minus g is nothing but the usual white noise term. Right. So, firstly, you should understand what terms you would have if you go on replacing the values of j. So, if j is 0, this would be e t. Then, if j is 1, this would be e t minus 1.

So, if j is 2, it will be e t minus 2, etc. So, in other words, this infinite sum would be nothing but something like psi 0, then e t plus psi 1 e t minus 1 plus psi 2 e t minus 2 plus etc. So, as you go on increasing the index j, you do not have any upper bound because you have an infinite sum, but then essentially this infinite sum reduces to this. So, whenever d happens to be between minus half and half, a stationary solution of the previous equation reduces to nothing but this. Now, immediately once you define the RFEMA or rather fractionally integrated noise, which is RFEMA 0D0, then one can talk about its autocorrelation function or, let us say, autocorrelation function, etc.

By the way, all these things could be proved, but then again we need not go into details here because we will try to keep it sort of simple. I mean, I will try to only define, let us say, whatever new processes we are kind of introducing and then rather than going into intricate, let us say, proofs and all of that. But immediately once you define the Arfima 0D0 process, I can talk about its autocovariance function. So, gamma k takes this form. Now, again here, if you see closely, this is predominantly dominated by, let us say, gamma functions. So, gamma 1 minus 2 d into gamma of k plus d, and then you have a certain denominator.

$$\gamma_k = \frac{\Gamma(1-2d)\Gamma(k+d)}{\Gamma(1-d)\Gamma(d)\Gamma(k-d+1)}\sigma_e^2$$

For $k = 0$,

$$\gamma_0 = \frac{\Gamma(1-2d)}{[\Gamma(1-d)]^2}\sigma_e^2$$

So, gamma 1 minus d into gamma d into gamma of k minus d plus 1 and then times this variance okay and what would happen if you plug in k to be 0 so k to be 0 is nothing but gamma 0 which is nothing but the variance of the process, right? So essentially the variance of any rthema 0 comma d comma 0 process reduces to this And again, you can clearly see that what would happen if you plug in k to be 0 here. So, let us say if k is 0, this would reduce to gamma d and then I can cancel out gamma d and then gamma d from here and so on and so forth. And again, I can replace k to be 0 here.

So, this would become gamma of 1 minus d. So, you have gamma of 1 minus d whole square in the denominator and multiplied by sigma square e here. So, this would be the autocorrelation function and then similarly I can talk about the ACF also autocorrelation function and so on. So, autocorrelation function is nothing but gamma k divided by the square root of the variances. So, rho k sort of reduces to that. But here we can sort of invoke a very interesting idea which is called a Stirling's formula.

So again I am not sure that how many of you are familiar with this approximation. But using the Stirling's approximation or the Stirling's formula I can have a very neat looking approximation for the gamma function. So gamma of x is nothing but under root 2 piinto e to the power minus x plus 1 into x minus 1 to the power x minus 1 half as of course, x goes to infinity. So, this is more like an approximation, right? But it sort of reduces the computational aspect of sort of let us say solving for rho k or gamma k because both gamma k and rho k have gamma functions right.

And then if you have a sterling approximation then the compute time reduces right. So, because rather than finding out factorial. So, if you have a very large number finding factorial becomes tedious right both mathematically as well as computationally right. Let us say let us say if you want to find out gamma of 100. So, gamma of 100 is nothing but 99 factorial right.

$$\rho_k = \frac{\Gamma(1-d)\Gamma(k+d)}{\Gamma(d)\Gamma(k-d+1)}$$

So, it it would be tedious in terms of mathematical and then also computationally. So, if you apply the Sterling's approximation then things are slightly easier ok. And then of course, both gamma k and rho k as we discussed earlier sort of predominate predominantly contain the gamma functions. So, why not simply replace the approximation whenever you have any gamma functions ok. So, if you apply the Stirling's approximation, my firstly my psi j because psi j again contains or is a combination of several gamma functions.

So, I can sort of apply the Stirling's approximation even here and my psi j coefficient reduces to this way. So, j to the power minus d minus 1 divided by gamma of minus d as j goes to infinity. And on the other hand, what would happen to my ACF? So, my ACF also reduces to something like this. So, gamma of 1 minus d divided by gamma of d and then into k to the power 2d minus 1 again as k goes to infinity, right.

But here the whole idea is, let us say we will talk about two conditions. So, for d very close to 0.5, right? Again, just to repeat, if d is, which is my operator or the differencing operator. So, if my D is very close to 0.5, then rho k decays very slowly, and hence the process has long memory. So, immediately we sort of get one characteristic of a particular value of D as to when the ARFIMA(0,D,0) would sort of become persistent or it would sort of preserve the long memory aspect. As opposed to that, if my D is sort of close to minus 0.5, then what happens?

$$\psi_j \sim \frac{\Gamma(j+d)}{\Gamma(j+1)\Gamma(d)} \sim \frac{j^{-d-1}}{\Gamma(-d)}, \text{ as } j \to \infty.$$

$$\rho_k = \frac{\Gamma(1-d)}{\Gamma(d)} k^{2d-1}, \text{ as } k \to \infty.$$

For $d \sim 0.5$, ρ_k decays very slowly and the process has long memory.

For $d \sim -0.5$, ρ_k decays faster.

Then my rho k decays faster. So, if d is close to 0.5, my rho k decays slowly, and the process has long memory. On the other hand, if my d is close to minus 0.5, my rho k decays much faster. And all these things again can be proved using some convergence ideas from mathematics and so on and so forth. So, the only idea is that you should look at how these two structures behave as, let us say, j goes to infinity or k goes to infinity.

So, what would happen if I fix the value of d to be, let us say, either 0.5 or minus 0.5? Now, again, a long memory process. So, a long memory process is nothing but a stationary process where my ACF structure or rho k structure is sort of approximated by this guy. So, c into k to the power 2d minus 1, where, of course, this constant c need not be 0 or is not equal to 0, and my d has to be strictly less than 0.5. Now, again, two situations.

$$\rho_k \sim Ck^{2d-1},\ C \neq 0,\ d < 0.5.$$

$$d < 0,\ \ \sum_{k=-\infty}^{\infty} |\rho_k| < \infty \Rightarrow \text{process is an intermediate memory process.}$$

$$0 < d < 0.5,\ \ \sum_{k=-\infty}^{\infty} |\rho_k| = \infty \Rightarrow \text{process is a long memory process}$$

So, what would happen if my d is less than 0? Because again, remember what values of d we took. So, we are kind of assuming the values of d to be between minus half and half, right? So, my d could be either positive or negative, right? So, firstly, if d is negative—let us say less than 0—then this infinite sum.

So, summation k going from minus infinity to infinity, the absolute value of my ACF function is finite, right? Which implies that the process is an intermediate memory process. So, in this case, the process does not contain any long memory. So, the process is an intermediate memory process. So, somewhere in between—neither long memory nor short memory.

As opposed to that, if my D is strictly positive and then strictly between 0 and 0.5, then the same infinite sum—so summation k going from minus infinity to infinity of the absolute value of ACF—equals infinity. And this indicates that the process is a long memory process. So, all these are sort of special cases. So, what happens if my D is negative, and what happens if my D is positive but between, let us say, 0 and 0.5, right? So, in one of the cases, I get an intermediate memory process.

In the other situation, I get a long memory process, all right. So, I think just to summarize. So, the RFEMA model structure is as follows. So, let us say 1 minus B to the power D. So, operated on yt equals let us say psi b yt plus theta b et.

So, this would be a general ARFIMA process now. So, we are slowly transitioning from let us say ARFIMA 0 d 0 to a general ARFIMA process where you have a certain AR order, you also have a certain MA order, etc. And again, the AR order and the MA order are specified by nothing but these coefficients, as we even saw earlier. So, the phi coefficient and the theta coefficient are not new for us. So, the phi coefficient characterizes the AR part of the model, and the theta coefficient characterizes the MA part of the model.

But again, these are the special cases. So, what happens if D is 0? So, if D is 0, it implies an ARMA model. What happens if D is 1? So, D equal to 1 implies an ARIMA model.

And again, rightly so, because if D is 0, there is no differencing essentially. So, if d is 0, this is 1, right? So, 1 minus b to the power 0 is 1. So, you only have yt equal to that, which is nothing but an ARMA process, right? But if d is 1, you have this 1 minus b to the power 1, which is 1 minus b. So, you are basically differencing, right?

Because 1 minus b operated on yt is nothing but yt minus yt minus 1, okay? And thirdly, what happens if my d is between 0 and 0.5? So, my d is positive, but strictly between, let us say, 0 and then less than 0.5. So, such situations imply, let us say, long memory models. So, stationary, but persistent.

So, the model is stationary, but still persistent. And what happens if my D exceeds 0.5? So, this would be a special case. So, what happens if D exceeds 0.5? So, D is positive, but D is bigger than or equal to 0.5.

So, such a situation implies non-stationary models with very strong persistence. So, as and when you go on increasing the value of d, the tendency of persistence also sort of improves. Not improves, but rather increases, right? Because here, if you see that if d is less than 0.5, it sort of indicates a stationary model, but persistent. But whenever d becomes above 0.5, so d is bigger than or equal to 0.5, it sort of implies non-stationary models with very, very strong persistence.

So, I guess this is the idea of the ARFIMA model structure. So, again, just to summarize. So, by picking certain values of d—so, let us say 0 or probably 1, or d is positive but less than 0.5, or d is negative but bigger than minus 0.5, right? Then what happens? So, under each and every situation, what happens in the fractionally integrated noise as well as the general ARFIMA structure, okay?

So, for an ARFIMA, let us say (0, D, 0) process, which is nothing but a fractionally integrated noise because here you do not have any AR structure and you do not have any MA structure, right? So, in that case, these are the implications, right? So, if D is less than 0, then a process is an intermediate memory process, and whenever D is bigger than 0 but less than 0.5, then since that infinite correlation structure happens to be infinity, in that case, the process is a long memory process. And we talked about the last slide for a general ARFIMA model structure. So, D equal to 0 implies an ARMA model, D equal to 1 implies an ARIMA model, D between 0 and 0.5 implies long memory models, right.

So, again, stationary but persistent, and then D bigger than 0.5 implies non-stationary models with strong persistence, alright. So, in the next lecture, we will talk about a particular exponent called the Hurst exponent, which would again point to this entire discussion about, let us say, persistence or long memory. But by looking at the value of that Hurst exponent, one can actually immediately find out if there is persistence in the underlying time series or not, right. So, in today's lecture, just to finish it off, we talked about the model structure of the ARFIMA. But in the coming lecture, which is the next one, we will talk about a particular way of finding if the underlying time series is sort of persistent or, let us say, anti-persistent or what, etc.

Thank you.