4.2 Linear Processes and Autoregressive Moving Averages Models

Ming Lu

2022-04-02

A time series is a linear process if it has the form

for all , where is a sequence of constants with , and is a (weak) white noise (uncorrelated random variables) with zero mean and variance ; in symbols . The condition is to ensure that the series in Eq. (4.3) converges. A particular case of Wold’s decomposition theorem asserts that any weakly stationary process has a linear representation of the form

where for all and , and is deterministic. Note that the linear process in the previous equation has , for all . This particular linear process is a (infinite order) moving average. From Wold’s decomposition of in Eq. (4.4), one can obtain the structure of autocovariances for in terms of the coefficients (the moving average weights), as follows:

From these equations one obtains the autocorrelations , and we will see that for particular cases of linear processes (such as the ARMA below), the deterministic component equals the mean times some constant. Thus, Wold’s representation is convenient for deducing properties of a stationary process. However, it is not of much use for modeling the behavior of the process, since estimation of from the infinite sums in Eq. (4.5) is not feasible. A class of models that can be seen as approximations to the Wold’s form of a stationary process and for which estimation is possible are the autoregressive and moving averages (ARMA) processes.

Definition (The class of ARMA processes) Consider a weak white noise, , and let integers and . A time series is (autoregressive of order ) if

MA (moving average of order if

(autoregressive and moving average of order if

where are real numbers, in all three equations.

Remark 4.1 We are considering the series with zero mean to simplify notation, but there is no loss of generality in this assumption, since if a process has mean , then fit the ARMA models to , which has zero mean. Note that this just adds a constant to the model. For example, suppose that follows an (1) process, so that , and this is equivalent to , where . If, on the other hand, follows a MA(1) process, then .

Any finite -order moving average (MA ) process is easily seen to be weakly stationary for any values of , since their autocovariances are finite versions of the sums in Eq. (4.5), and independent of :

Thus, a time series can be recognized as a process if its ACF is non-zero up to lag and is zero afterwards. In use the function acf to estimate autocovariances and autocorrelations (cf. Example 3.2). An infinite order moving average process is the special case of the general linear process in Eq. (4.3), with for all . It is stationary provided , and its autocovariance function is given by Eq. (4.5).

R Example 4.1 A natural finite order moving average process arises when one considers continuously compounded -period returns, for a given . Recall from Def. that for a given series of continuously compounded returns and a given time period , the -period return is expressed as a sum of log returns:

Assuming that is white noise, more precisely, iid random variables with , then the second order moments of satisfy

and . This shows that the structure of finite moments of is like a MA process, and so it can be simulated as such a type of processes:

with . One can observe this empirically in R. Build the series of daily log returns from the price history of your favorite stock; check that there are no significant autocorrelations for any lag (using the acf function, or the Ljung-Box statistics-see Eq. (3.6)); build a series of -period returns, for say (a monthly period), and compute the ACF of this series. An important lesson is learned from this example, and that is while might be white noise (iid random variables), the -period series is not, and in fact has a smoother behavior.

Now, to deal with the AR and ARMA models, it is better to rewrite their representation in a compact form using lag operators. Lag operators and polynomials. The lag operator, or backward shift, acts on a time series by moving the time index back one time unit:

Note that , and in general, the -power of moves to its lag : . We can then define the polynomials in the lag operator,

With these polynomials in , we can express the autoregressive and moving averages models in the following succinct forms:

These representations allow to transform an process into a MA( , and a into an , from where one can deduce conditions on the coefficients implying the stationarity of the AR (and ARMA) processes. Going from AR to MA. The lag polynomials expressions suggest that to transform an process into a MA we just have to multiply by the inverse polynomial , so that, . But what is the inverse of (if it exists)? Consider and AR(1) process

To invert , recall the Taylor polynomial expansion of :

defined for . This fact suggests that

provided . Although this is an informal argument (we have not said much about the operator , and are assuming that implies ), one can verify that the obtained MA( process, , is the (unique) stationary solution of (4.13). Thus, the zero mean process (4.13) is stationary if , in which case it has the moving average (or Wold) representation given by Eq. (4.14). Autocovariances and autocorrelations of an . From Eq. (4.14), one can compute

Autocovariances and autocorrelations of an . From Eq. (4.14), one can compute

and the autocorrelations, and . This last equation says that the ACF of an AR(1) process decays exponentially with rate and starting value . Thus observing an inverse exponential curve in the plot of the (sample) ACF of a given time series is a sign of the existence of an underlying AR(1) process.

There are many financial time series that can be well modeled by an process. Examples are interest rates, real exchange rates, and various valuation ratios, such as price-to-earning, dividend-to-price and others that will be studied in Chap.

Autocovariances and autocorrelations of an . From Eq. (4.14), one can compute

and the autocorrelations, and . This last equation says that the ACF of an AR(1) process decays exponentially with rate and starting value . Thus observing an inverse exponential curve in the plot of the (sample) ACF of a given time series is a sign of the existence of an underlying AR(1) process.

There are many financial time series that can be well modeled by an process. Examples are interest rates, real exchange rates, and various valuation ratios, such as price-to-earning, dividend-to-price and others that will be studied in Chap.

Now, in order to apply a similar idea for transforming an process, given in the form of Eq. (4.10), into a MA( process, factor the lag polynomial into a product of linear factors

where the can be complex numbers. Assuming that are all different, we have

where the coefficients are obtained by a partial fractions decomposition, and depend on . If we assume that for , we can write each factor . Then, the process

is a stationary process, provided . This last condition can be shown to be equivalent to

Thus, the process (4.10) is stationary if Eq. (4.18) holds, and a solution has the general form (4.17). It will be very instructive for the reader to work out the details for the process, and obtain the explicit values for and in terms of and , see Problem 4.7.2.

Remark 4.2 If there are repeated factors of degree in Eq. (4.16), then in the partial fraction decomposition appear terms . Each term , for , is the derivative of order of times some constant; hence taking derivatives to the Taylor expansion of we get the desired series for the powers of this fraction; specifically .

Autocovariances of an and Yule-Walker equations. The structure of the autocovariance function for a process , is not as easy to get directly from its MA( (or Wold) expression (Eq. (4.17)) as in the AR(1) case. Here the trick is to consider the process in its original form

and note that since , the autocovariances . Then multiplying each side of (4.19) by , taking expectations and using that , where (Eq. (4.17)), to evaluate the right-hand sides (i.e., to compute ), we obtain the following system of equations for the moments of the process

These are the Yule-Walker equations. In matrix form,

where and is the autocovariance matrix (recall from Remark that ).

The Yule-Walker equations are a useful tool to estimate the coefficients of an model to fit to a given observed process . Consider the sample autocovariance,

for . Then, we have that the sample covariance matrix

is invertible, and from Eq. (4.21) we have

The solution is an unbiased estimation of provided the sample size is taken large enough, and . A rough guide is to take and (Box and Jenkins 1976).

Forecasting with AR . The reader should have noticed the similarities of the Yule-Walker equations with the linear equations (2.45) in Sect. verified by the coefficients of a one-step linear forecaster. The procedures to derive both systems of equations are exactly the same. Indeed, if we consider an process

the coefficients satisfy the Yule-Walker equations (4.20) with variables, which can be written as

On the other hand, the best linear forecaster of in terms of is

where satisfy the linear equations (2.45):

By the unicity of the solutions to these equations, it must be

Therefore, for an process, the best linear forecaster of in terms of is

(i.e., it is determined by the same coefficients that define the autoregressive process). The mean square error of the prediction, , is given by Eq. (2.47). The particular case of forecasting with an process is quite simple. The best linear one-step forecaster of an AR(1) process (where and in terms of its past is

and using Eq. (4.15) we can express the mean square error of the prediction as

A rule to determine the order . What remains to know is a method to determine the order of the model. Just as in an process, a plot of the ACF function will show a sinuous shape, a mixture of sines, cosines and exponential decay, and this gives an indication that an AR model of certain order is possible. But to actually determine the order of the autoregressive model use the partial autocorrelation function (PACF), which is based on setting up the following AR models listed in increasing orders:

and solve this multiple linear regression by the least squares method. Then the rule for fitting an model is that the first coefficients are non-zero and for the rest . The PACF criterion is programmed in with the function pacf (and we have already use it in R Example 4.1). The model estimation based on the Yule-Walker equations (Eq. (4.22)) is done in R with the function . Example (Continue from Example 4.1) The EUR/USD exchange rates behave as an AR(1) process: its ACF function showed an exponential decay and

Going from MA to AR. Applying similar ideas as the transformation of a finite order process into an infinite order MA, we can go from a process to process by inverting the polynomial in Eq. (4.11), so that becomes an autoregressive process of infinite order. The polynomial is invertible if holds for all . The MA process is invertible if the polynomial is invertible. Let us illustrate this transformation explicitly for the case of moving averages of order Example 4.2 We are given a MA(1) process in lag operator form,

Using the Taylor expansion for the function , which is defined for as follows

we obtain

or equivalently, , which is clearly an process.

What we can observe from this representation is that the PACF for an invertible MA process has an exponential decay towards zero. ARMA autocovariances and model estimation. From the back-and-forth transformations between AR and MA process, what we can readily say about the autocorrelation functions of an process is that for the ACF behaves similarly to the ACF of an process, while for the PACF behaves like the PACF of a MA process, and, consequently, both ACF and PACF of the ARMA process decay exponentially for large and . To compute an instrumental expression for the of an process we proceed with an analysis similar to the AR case (i.e., a derivation like Yule-Walker’s). Consider an process with zero mean or mean-corrected (i.e., the mean has been subtracted from the series), written in the form

with . Multiply both sides of this equation by , for , and take expectations. Using the Wold representation, , and that since , we obtain from the previous operations the following equations:

and

where for , and for . Assuming that the lag polynomial, , has distinct roots, then the system of homogeneous linear difference Eq. (4.24) has a solution of the form

where are the distinct solutions of , and are arbitrary constants. Since we want a solution of Eq. (4.24) that also satisfies Eq. (4.23), substitute the general solution (4.25) into Eq. (4.23) to obtain a set of linear equations that uniquely determine the constants , the autocovariances , for , and consequently, the autocorrelations . The procedure just described to compute the autocorrelations of an process is implemented by the function ARMAacf of the stats package in R.

Now, to fit an ARMA model to a given zero-mean time series one first chooses a desired order pair , and assuming that the observed data is stationary and Gaussian one uses the maximum likelihood estimation (MLE) method with respect to parameters , , to find estimators and . These maximum likelihood estimators can be obtained as solutions to the following system of equations:

where

and are the values of and that minimize

where is the size of the sample observations, and , for , are the one-step predictors; the are obtained from the covariances of and are independent of is a “reduced log-likelihood”. The derivation of this system of equations is not difficult but involved, and can be studied in full details in Brockwell and Davis (1991, Chap. 8). The MLE for ARMA is a nonlinear optimization problem and so it requires a numerical procedure to iteratively find the values of the parameters that minimized the square prediction error. It needs to be given some initial values for the parameters to start, and convergence and correctness will depend to some extend on these initial values. Hence, some appropriate approximation methods should be employed to find good initial values. Two such initialization methods are the innovation algorithm and Hannan-Rissanen algorithm (again see Brockwell and Davis op. cit.).

Finally, to select appropriate values for the orders and one uses the Akaike information criterion (AIC), which consists on choosing and that minimize

where is the Gaussian likelihood for an process. Thus, in practice the AIC criterion works as follows: choose some upper bounds, and , for the AR and MA order respectively. Then fit all possible models for and , on the same sample series of size , and keep the model whose parameters and minimize AIC. Beware though that this criterion asymptotically overestimates the order, that is, as larger values of and are chosen. It can also be a time-consuming optimization procedure for high orders and . However, the general wisdom asserts that in financial applications small values of and (say, less than 5) are sufficient. In the following Example we program the AIC criterion and use it to fit an ARMA model to financial data. Example 4.4 We consider adjusting an ARMA model to Robert Shiller’s Priceto-Earning (P/E) ratio 5 for the S&P 500. This series, also known as Shiller’s PE 10 , is based on the average inflation-adjusted earnings of the S&P 500 taken over the past 10 years, and computed every month since 1881 . The data is publicly available from Shiller’s webpage, and we’ve saved it in our disk with the label SP500\_shiller.csv. We implement the AIC criterion by setting up a double loop to apply for all pairs of , from to upper bound in , the armaFit method on the data, and keep the model with minimum AIC. To extract the AIC value computed in each execution of armaFit one uses the expression [tsARMA@f](mailto:tsARMA@f) it $aic, where tsARMA is the name of the variable that saves the current computed ARMA. Because the method armaFit might fail to compute a model, in which case it returns an error or warning message producing an interruption of the loop, we execute armaFit within tryCatch to handle these error messages. The full program follows.