STAT 153 & 248 - Time Series
Lecture Twenty Two

Fall 2025, UC Berkeley

Aditya Guntuboyina

November 18, 2025

1 ARMA(p, q) models

The ARMA(p, q) model is defined by the equation:

$$(y_t - \mu) - \phi_1 (y_{t-1} - \mu) - \cdots - \phi_p (y_{t-p} - \mu) = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} \tag{1}$$

where, as usual, $\epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$. The unknown parameters in this model are $\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$ and the noise standard deviation σ.

An equivalent way of writing (1) is by moving the μ terms to the right hand side. This gives:

$$y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p} = \mu (1 - \phi_1 - \cdots - \phi_p) + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}. \tag{2}$$

In other words,

$$y_t = \phi_0 + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \eta_t \tag{3}$$

where $\phi_0 = \mu(1 - \phi_1 - \cdots - \phi_p)$ and $\eta_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$. In this sense, ARMA($p$, q) can be seen as an AR(p) model where the error terms are modeled as MA(q) instead of i.i.d Gaussian $N(0, \sigma^2)$.

AR(p) and MA(q) models are special cases of ARMA(p, q) because:

1. When $p = 0$, the equation (1) becomes:

   $$y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

   which is the MA(q) model. So ARMA(0, q) = MA(q).

2. When $q = 0$, the equation (2) becomes:

   $$y_t = \phi_0 + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \epsilon_t$$

   which is the AR(p) equation with $\phi_0 = \mu(1 - \phi_1 - \cdots - \phi_p)$. Thus ARMA($p$, 0) = AR($p$).

In Backshift notation, we can write

$$\phi(B) (y_t - \mu) = \theta(B) \epsilon_t \tag{4}$$

where $\phi(B)$ and $\theta(B)$ are the AR and MA polynomials:

$$\phi(z) := 1 - \phi_1 z - \cdots - \phi_p z^p \quad \text{and} \quad \theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$$

applied to the Backshift operator B.

We can rewrite (4) as

$$y_t - \mu = \frac{\theta(B)}{\phi(B)} \epsilon_t.$$

To make sense of the right hand side above, we can factorize $\phi(z)$ as:

$$\phi(z) = (1 - a_1 z)(1 - a_2 z) \ldots (1 - a_p z).$$

Here $1/a_1, \ldots, 1/a_p$ are the roots of $\phi(z)$. This gives

$$y_t - \mu = \frac{\theta(B)}{\prod_{k=1}^{p}(1 - a_k B)} \epsilon_t = \theta(B)(1 - a_1 B)^{-1} \ldots (1 - a_p B)^{-1} \epsilon_t$$

Suppose each $|a_k| < 1$ (in other words, every root of $\phi(z)$ has modulus strictly larger than 1), then each term $(1 - a_k B)^{-1}$ can be expanded as follows:

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

This allows us to write $y_t - \mu$ in terms of $\{\epsilon_t\}$:

$$y_t = \mu + \sum_{j=0}^{\infty} \psi_j \epsilon_{t-j}$$

for some $\psi_0, \psi_1, \psi_2, \ldots$. This is a causal stationary process. We shall only work with ARMA(p, q) models in the causal stationary regime (which corresponds to $\phi(z)$ having all roots of modulus strictly larger than 1).

Therefor if the AR polynomial $\phi(z)$ has all roots with modulus strictly larger than 1, then the ARMA(p, q) difference equation has a stationary and causal solution:

$$y_t = \mu + \psi_0 \epsilon_t + \psi_1 \epsilon_{t-1} + \psi_2 \epsilon_{t-2} + \ldots.$$

These coefficients $\{\psi_j\}$ can be explicitly determined in terms of $\phi_1, \ldots, \phi_p$ and $\theta_1, \ldots, \theta_q$ by solving the equation: $\psi(z) = \theta(z)/\phi(z)$ (here $\psi(z) := \psi_0 + \psi_1 z + \psi_2 z^2 + \ldots$) which can be done by writing

$$\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q = \phi(z) \times \psi(z) = (1 - \phi_1 z - \cdots - \phi_p z^p)\left(\psi_0 + \psi_1 z + \psi_2 z^2 + \ldots\right)$$

and then equating the coefficients of z^j on both sides for $j = 0, 1, \ldots$ to get

$$1 = \psi_0, \quad \theta_1 = \psi_1 - \psi_0 \phi_1, \quad \theta_2 = \psi_2 - \psi_1 \phi_1 - \psi_0 \phi_2, \quad \theta_3 = \psi_3 - \phi_1 \psi_2 - \phi_2 \psi_1 - \phi_3 \psi_0, \quad \ldots$$

Note that the condition for ARMA(p, q) to have a causal stationary solution is identical to the condition for AR(p) to have a causal stationary solution (namely, every root of the polynomial $\phi(z)$ should have modulus strictly larger than 1).

2 ACF and PACF of ARMA(p, q) models

For a causal stationary time series model, we can define ACF and PACF. The ACF is simply defined as:

$$ACF(h) = \text{correlation between } y_t \text{ and } y_{t+h}.$$

The PACF is defined as (it can be shown that the quantity below does not depend on t if $\{y_t\}$ is stationary):

$PACF(h) =$ partial correlation between y_t and y_{t+h} after removing the effect of $y_{t+1}, \ldots, y_{t+h-1}$.

We will not go into the formal definition of partial correlation between two random variables X_1 and X_2 after removing the effect of $Z_1, \ldots, Z_m$ (see e.g., `https://en.wikipedia.org/wiki/Partial_correlation`).

The key facts are:

1. For the MA(q) model, $ACF(h)$ is exactly zero for $h > q$. So ACF(h) can be used to determine the value of q for the MA model.
2. For the AR(p) model, $PACF(h)$ is exactly zero for $h > p$. So PACF(h) can be used to determine the value of p for the AR model.
3. For an ARMA(p, q) model with both $p \geq 1$ and $q \geq 1$, neither the ACF nor the PACF cuts off after a certain lag. For such models, ACF and PACF are not useful for determining p and q. There are inbuilt functions `arma_acf` and `arma_pacf` for computing the ACF and PACF of causal stationary ARMA processes. These can be used to get an idea of the behaviour of the ACF and PACF for ARMA processes.
4. Given observed data $y_1, \ldots, y_n$ that is supposedly generated from a stationary time series model, we can estimate $ACF(h)$ and $PACF(h)$ for every h. These estimates are called Sample ACF and Sample PACF respectively. These can be obtained from inbuilt functions in `statsmodels`. These are useful for determining q (in order to use an MA(q) model) or p (in order to use an AR(p) model). But they are not useful for determining both p and q in order to use an ARMA(p, q) model.

Instead of using sample ACF and PACF, we use automatic model selection criteria such as AIC and BIC to determine p and q while using ARMA(p, q) models.

3 Parameter Estimation, and AIC and BIC

For parameter estimation in ARMA(p, q) models, we simply use the `ARIMA` function from `statsmodels`. This uses maximum likelihood estimation. Writing the likelihood for ARMA models is not easy, and the function uses some ideas from state space modeling (and the Kalman filter) for writing the likelihood. We will not go into the details of this. It should be noted that, quite often, the `arima` function will give warnings and errors while fitting ARMA models. We will be ignoring these messages.

The AIC and BIC for a fitted ARMA(p, q) model are calculated as:

$$AIC = (-2) \times \text{maximized log-likelihood} + 2 \times \text{number of parameters}$$

$$BIC = (-2) \times \text{maximized log-likelihood} + (\log n) \times \text{number of parameters}$$

These can be used for model selection. Models with smaller values of AIC and BIC are preferred. The first term in the definition of AIC and BIC determines the quality of fit to the data, and the second term penalizes model complexity thereby guarding against overfitting.

Generally $\log n$ will be larger than 2 so BIC will lead to sparser models (i.e., models with fewer number of parameters) compared to AIC.

4 The Box-Jenkins Time Series Modeling Strategy

Box and Jenkins popularized the following strategy for modeling an observed time series $y_1, \ldots, y_n$:

1. Generally $y_1, \ldots, y_n$ will exhibit various kinds of trends. Preprocess the data to transform it to another series x_t which does not have any discernible trends.
2. Fit an ARMA(p, q) model for appropriate p and q to the transformed data x_t.

The preprocessing in the first step above is usually done by taking differences (either for the original data or for its logarithms). The first difference of $\{y_t\}$ is given by $\nabla y_t := y_t - y_{t-1}$ for $t = 2, \ldots, n$. The second difference is given by

$$\begin{aligned}\nabla^2 y_t &= \nabla(\nabla y_t) \\ &= \nabla(y_t - y_{t-1}) = \nabla y_t - \nabla y_{t-1} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}.\end{aligned}$$

Higher order differences $\nabla^k y_t$ are defined recursively. Note that the length of the time series comes down after each successive differencing. For example, ∇y_t has length $n-1$, $\nabla^2 y_t$ has length $n-2$ and so on. Differencing usually eliminates increasing/decreasing trends. Usually one or two orders of differencing is enough to take care of increasing/decreasing trends.

To the transformed data x_t, one fits an ARMA(p, q) model which can be done via the `ARIMA` function from the statsmodels library. The order p and q can be determined via a model selection criterion such as AIC or BIC.

5 ARIMA models

ARIMA stands for AutoRegressive Integrated Moving Average. ARIMA is essentially differencing plus ARMA.

Definition 5.1 (ARIMA)**.** *A time series model y_t is said to be ARIMA(p, d, q) if*

$$\phi(B)((\nabla^d y_t) - \mu) = \theta(B)\epsilon_t,$$

where $\epsilon_t \overset{i.i.d}{\sim} N(0, \sigma^2)$.

ARIMA models are fit by the function `ARIMA()` in `statsmodels`. The mean μ above is taken to be zero by default when the order parameter d in `ARIMA` is strictly larger than zero.

When $d = 1$, the ARIMA(p, 1, q) model becomes:

$$\phi(B)(\nabla y_t - \mu) = \theta(B)\epsilon_t. \tag{5}$$

Define a process η_t by $y_t = y_0 + \mu t + \eta_t$ (in other words, $\eta_t = y_t - y_0 - \mu t$). Then

$$\nabla y_t = y_t - y_{t-1} = (y_0 + \mu t + \eta_t) - (y_0 + \mu(t-1) + \eta_{t-1}) = \mu + (\eta_t - \eta_{t-1}) = \mu + \nabla\eta_t.$$

Plugging this into (5), we get

$$\phi(B)\nabla\eta_t = \theta(B)\epsilon_t.$$

In other words, (5) is equivalent to:

$$y_t = y_0 + \mu t + \eta_t \qquad \text{where } (\nabla\eta_t) \text{ is ARMA}(p,\, q) \text{ with zero mean.}$$

When $\mu \neq 0$, the term $y_0 + \mu t$ represents a deterministic linear trend in y_t. However many real-world time series (e.g., macroeconomic variables such as GNP or GDP) are unlikely to exhibit an exact deterministic linear trend. For this reason, the default behavior of the `ARIMA` function in `statsmodels` sets $\mu = 0$ in (5). To fit it with $\mu \neq 0$, one must specify the argument `trend = 't'`.