CHAPTER 2

Time Domain Methods

Time domain methods do not employ any form of transform space to describe a time series (although it is commonly the case that one can best understand their structures by analyzing them in the frequency domain). The names most associated with these techniques are Wiener, and Box and Jenkins. As with the frequency domain methods, one can begin the discussion in continuous time and it was one of Wiener's great contributions to show how to deal with that case. But continuous time representations raise all sorts of complex mathematical issues that disappear when a time series is made discrete, and so for present purposes, we will begin with the discrete case of a uniformly sampled time series x_t.

1. Representations-1

As with Fourier methods, much of the purpose of these methods is to find efficient representations of stochastic processes whose interpretation can lead to physical insights. For notational simplicity, we will assume that $\Delta t = 1$. Consider the simple rule (actually a difference equation of similar form to (7.1) above),

$$x_{m+1} = ax_m + \theta_m \tag{1.1}$$

where a is a constant and θ_m is a zero-mean white noise process of variance σ_θ^2. Starting with $x_0 = 0$, (1.1) permits simple generation of realizations of x_m depending upon the particular run of random numbers θ_m (Fig. 28). We can compute the autocovariance of x_m :

$$R(0) =< x_m^2 >=< (ax_{m-1} + \theta_{m-1})^2 >= a^2 R(0) + \sigma_\theta^2 \tag{1.2}$$

where we used $< x_{m-1}\theta_{m-1} >= 0$, and the assumption that the time-series was wide-sense stationary ($< x_{m-1}^2 >=< x_m^2 >= R(0)$). So,

$$R(0) = \frac{\sigma_\theta^2}{(1-a^2)}. \tag{1.3}$$

Evidently, there would be a problem if $a = 1$, and in fact, $|a| < 1$ proves to be necessary for the time-series to be stationary. Similarly,

$$R(1) =< x_{m+1}x_m >=< (ax_m + \theta_{m+1})\, x_m >= aR(0). \tag{1.4}$$

Exercise. Find $R(2), ..., R(m)$ for x_t in (1.1).

If one knew $R(0)$ and $R(1)$, Eqs. (1.3, 1.4) would fully determine a, σ_θ^2 and they in turn fully determine everything there is to know about it. Before asking how one might determine $R(0), R(1)$, let us ask where an equation such as (1.1) might arise?

Consider a simple differential system

$$\frac{dx(t)}{dt} = Ax(t) + g(t) \tag{1.5}$$

where A is constant and θ is any externally imposed forcing. Equations like this are used to describe, e.g., a local change in a heat content anomaly, $x(t)$, as the result of conduction from a reservoir, heat loss by radiation, and external sources g. Forming simple one-sided time differences, (1.5) becomes

$$x(m\Delta t + \Delta t) = \Delta t(A+1)x(m\Delta t) + \Delta t g(m\Delta t) \tag{1.6}$$

or,

$$x_{m+1} = \Delta t(A+1)x_m + \Delta t g_m \tag{1.7}$$

which is of the form (1.1) with $a = \Delta t(A+1)$. Two types of problem exist. In one, g_m is known, and one seeks a; in the other type, $g_m = \theta_m$ is unknown and believed to be a white noise process..

In the second type of problem one has observations of x_t and the question is what the best estimates of a, σ_θ^2 are. Let us try least-squares by minimizing,

$$J = \sum_{m=0}^{N-1} (x_{m+1} - ax_m)^2. \tag{1.8}$$

The argument here would be that (1.1) can be regarded as an equation which forecasts x_{m+1} from x_m, and minimizing the unpredictable part, θ_m, would give the best possible forecast system. The normal equations for (1.8) are just one equation in one unknown,

$$a\sum_{m=0}^{N-1} x_m^2 = \sum_{m=0}^{N-2} x_{m+1}x_m. \tag{1.9}$$

Divide both sides of this equation by N,and we see that it can be written as

$$a\tilde{R}(0) = \tilde{R}(1), \tag{1.10}$$

where we recognize

$$\frac{1}{N}\sum_{m=0}^{N-1} x_m^2, \tag{1.11}$$

as an *estimate* of the true autocovariance $R(0)$, and similarly for $R(1)$. Given the resulting estimate of a, call it $\tilde{a}$, one can substitute into (1.8) and compute the estimate $\tilde{\sigma}_\theta^2$.

A more general form of the representation of a time-series is,

$$x_{m+1} = a_1 x_m + a_2 x_{m-1} + ... + a_M x_{m-M+1} + \theta_{m+1}, \tag{1.12}$$

which is called an "autoregressive process of order M" or AR(M), so that (1.1) is an AR(1) process. To determine the coefficients a_i we can proceed again by least-squares, to find the minimum of

$$J = \sum_{m=0}^{N-1} \left(x_{m+1} - a_1 x_m - a_2 x_{m-1} - ... - a_M x_{m-M+1} \right)^2 \tag{1.13}$$

and forming the normal equations,

$$\begin{aligned} a_1 \tilde{R}(0) + a_2 \tilde{R}(1) + a_3 \tilde{R}(2) + .. + a_M \tilde{R}(M-1) &= \tilde{R}(1) \\ a_1 \tilde{R}(1) + a_2 \tilde{R}(0) + a_3 \tilde{R}(1) + .. + a_M \tilde{R}(M-2) &= \tilde{R}(2) \\ &... \\ a_1 \tilde{R}(M-1) + a_1 \tilde{R}(M-2) + a_3 \tilde{R}(M-3) . + . + a_M \tilde{R}(0) &= \tilde{R}(M) \end{aligned} \tag{1.14}$$

where we used $\tilde{R}(-k) = \tilde{R}(k)$. Equations (1.14) are usually known as the Yule-Walker equations. Solving them produces an estimate of the vector of unknowns $\mathbf{a} = [a_1, ...a_M]^T$ and the value of J is the estimate of σ_θ^2. If (1.14) is written in matrix form

$$\tilde{\mathbf{R}}\mathbf{a} = \mathbf{b} \tag{1.15}$$

one sees that $\tilde{\mathbf{R}}$ is a covariance matrix having the special property that all diagonals have the same values:

$$\tilde{\mathbf{R}} = \begin{Bmatrix} \tilde{R}(0) & \tilde{R}(1) & \tilde{R}(2) & . & \tilde{R}(M-1) \\ \tilde{R}(1) & \tilde{R}(0) & \tilde{R}(1) & . & \tilde{R}(M-2) \\ \tilde{R}(2) & \tilde{R}(1) & \tilde{R}(0) & . & \tilde{R}(M-3) \\ . & . & . & . & . \\ \tilde{R}(m-1) & \tilde{R}(m-2) & \tilde{R}(m-3) & . & \tilde{R}(0) \end{Bmatrix} \tag{1.16}$$

A matrix with constant diagonals is called "Toeplitz", and the special form of (1.15) permits the system of equations to be solved without a matrix inversion, using an extremely fast recursive algorithm called the Levinson (or sometimes, Levinson-Derber) algorithm. This possibility is less important today than it was in the days before fast computers, but if M is extremely large, or very large numbers of systems have to be solved, the possibility can remain important.

If g_m is a known time-series, one can proceed analogously by minimizing via least-squares, the objective function

$$J = \sum_{m=0}^{N-1} \left(x_{m+1} - a x_m - g_m \right)^2 \tag{1.17}$$

with respect to a. Higher order generalizations are obvious, and details are left to the reader.