# 5: SMOOTHING, LINEAR FILTERING

Suppose we have data $y_t = a_t + e_t$ where $\{a_t\}$ is a smooth function (the "signal"), and $\{e_t\}$ consists of random errors (the "noise"). **Smoothing** refers to the process of trying to (approximately) recover $a_t$ from the data $y_t$. Smoothing is traditionally carried out by local averaging of the data $y_t$, an operation called **linear filtering.** For example, we might average $y_{t-1}, y_t$ and $y_{t+1}$ to form

$$z_t = \frac{1}{3}[y_{t-1} + y_t + y_{t+1}] = \frac{1}{3}[a_{t-1} + a_t + a_{t+1}] + \frac{1}{3}[e_{t-1} + e_t + e_{t+1}] \quad .$$

Since $\{a_t\}$ is smooth, the first term on the RHS is approximately $a_t$. Since the errors are "random", they tend to cancel out so that the second term on the RHS will tend to be smaller than the individual errors. Thus, this linear filter can help us in extracting the smooth series $\{a_t\}$ from $y_t$. In general, a linear filter is determined by a set of weights $\{g_r, g_{r+1}, \ldots, g_s\}$ such that if the input to the filter is $\{y_t\}$, the output is

$$z_t = \sum_{u=r}^{s} g_u y_{t-u} \quad .$$

We say that $z_t$ is the result of passing $y_t$ through a linear filter with weights $g_r, \ldots, g_s$. Note that $z_t$ is a local average of $y_t$. $z_t$ can also be described as a **convolution** of the data sequence $\{y_t\}$ with the weight sequence $\{g_u\}$. Linear filters have two important properties. First, they are **invariant,** i.e., their properties do not change with time. A given time shift in the input will simply produce the same shift in the output. Second, linear filters are **superposable:** If the input is the sum of two series, the output will be the sum of the two outputs produced by the individual series.

The effects of a linear filter are best thought of in terms of the frequency domain. The key question is: what happens to the complex exponential $y_t = \exp(i\omega t)$ when we pass it through the linear filter? The answer is

$$z_t = \sum_{u=r}^{s} g_u \exp(i\omega(t-u)) = \exp(i\omega t)\sum_{u=r}^{s} g_u \exp(-i\omega u) = \exp(i\omega t)\, G(\omega) \quad ,$$

where

$$G(\omega) = \sum_{u=r}^{s} g_u \exp(-i\omega u) \quad .$$

Thus, the output is still an exponential at frequency $\omega$, but it is now multiplied by the constant (i.e., time-independent quantity) $G(\omega)$. $G(\omega)$ is called the **transfer function** since it describes how the exponential at frequency $\omega$ is transferred from the input to the output. Since any data sequence can be written as a linear combination of complex exponentials, the transfer function $G(\omega)$ tells us in general how the *component* of our data at frequency $\omega$ is transferred from input to output. We will make this statement more precise in the following paragraph. For the example considered earlier, $g_{-1}=g_0=g_1=\frac{1}{3}$, and

$$G(\omega) = \frac{1}{3}[\exp(i\,\omega) + 1 + \exp(-i\,\omega)] = \frac{1}{3}[1 + 2\cos\omega] \quad .$$

This function is maximized at $\omega=0$. Thus, low frequency components (which are dominated by the signal) will be passed almost unchanged, while high frequency components (which are dominated by the noise) will be cut down, or attenuated.

Linear filtering is equivalent to convolving the input sequence $y_t$ with the weight sequence $g_u$. Thus, if $\omega$ is a Fourier frequency, we have from Exercise 5.4 that the DFT of the output is

$$J_z(\omega) = n\, J_y(\omega)\, J_g(\omega) = G(\omega)\, J_y(\omega) \quad . \tag{1}$$

We see that linear filters do not move frequencies or create new ones; they simply change the phase and amplitude of existing harmonic components. The relation (1) holds for any input sequence, but it is only exactly true if a *circular* convolution is used, i.e., the index of $y$ is interpreted modulo $n$. The relation is still approximately true, however, under any reasonable definition for the unavailable end values of $y_t$ (namely $y_{-1}, y_{-2}, \ldots,$ and $y_n, y_{n+1}, \ldots$) which are needed for calculating $z_t$. Thus one method of computing the output $z_t$ from the input $y_t$ is:

1) Get the DFT of $\{y_t\}$ at the Fourier frequencies.

2) Multiply the DFT values $J_j$ by the transfer function $G(\omega_j)$.

3) Take the inverse Fourier transform of the result.