Chapter 7

Waveform applications of least squares

By the methods of calculus, one learns to find the coordinates of an extremal *point* on a curve. In the calculus of variations, one learns how to find extremal *functions*. In practice, the continuum may be approximated on a mesh and the distinction blurs. In the calculus of variations problems, however, the matrices can be immense, a disadvantage often partially offset by their orderly form. In this chapter we will take up examples in the use of least squares on waveforms and relationships between groups of waveforms. This leads to a massive full matrix called the block-Toeplitz matrix for which we have a special solution technique.

7.1 PREDICTION AND SHAPING FILTERS

A data wavelet is given by $\mathbf{b} = (b_0, b_1, \ldots, b_n)$. We plan to construct a filter $\mathbf{f} = (f_0, f_1, \ldots, f_n)$. Filtering is defined in this way: When data $\mathbf{b}$ go into a filter $\mathbf{f}$, an output wavelet $\mathbf{c}$ is produced according to the following matrix multiplication.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_{n+m} \end{bmatrix} = \begin{bmatrix} b_0 & 0 & \cdots & 0 \\ b_1 & b_0 & 0 & \\ b_2 & b_1 & b_0 & \\ \vdots & b_2 & b_1 & \ddots \\ b_n & \vdots & b_2 & \\ 0 & b_n & \vdots & \\ \vdots & & & b_{n-1} \\ 0 & 0 & \cdots & b_n \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_m \end{bmatrix} \tag{7.1}$$

This operation is often called *complete transient convolution*. It is the same as identifying coefficients in a polynomial multiplication.

Now we introduce another wavelet $\mathbf{d}$ which will have the same number of components as $\mathbf{c}$. We call $\mathbf{d}$ the desired output of the filter. We saw that $\mathbf{c}$ is the actual output. The actual output $\mathbf{c}$ was seen to be a function of the input $\mathbf{b}$ and the filter $\mathbf{f}$. The problem now is to determine $\mathbf{f}$ so that $\mathbf{c}$ and $\mathbf{d}$ are very much alike. Specifically we will choose $\mathbf{f}$ so that the difference vector $\mathbf{c} - \mathbf{d}$ has minimum length squared (in $n + m + 1$ dimensional space). In other words, we use the method of least squares to solve the overdetermined equations

$$\begin{bmatrix} b_0 & 0 & \cdots & 0 \\ b_1 & b_0 & 0 & \\ b_2 & b_1 & b_0 & \\ \vdots & b_2 & b_1 & \ddots \\ b_n & \vdots & b_2 & \\ 0 & b_n & \vdots & \\ \vdots & & & b_{n-1} \\ 0 & 0 & \cdots & b_n \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_m \end{bmatrix} \approx \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d_{n+m} \end{bmatrix} \tag{7.2}$$

Using the "quick-and-dirty" method of the previous chapter we merely premultiply (7.2) by the transposed matrix. The result is a Toeplitz matrix of the form

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_m \\ r_1 & r_0 & r_1 & & \\ r_2 & r_1 & r_0 & & \\ \vdots & & & \ddots & \\ r_m & & & & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} \tag{7.3}$$

where r_k is the autocorrelation of the input x_k and g_k is a crosscorrelation of the input x_k with the desired output d_k. For computation techniques see Chapter 7.5.

The formulas of this section may also be used to attempt to predict a time series from its past. For example $f_1, f_2, \ldots, f_m$ is a prediction filter of x_{t+10} from $x_t, x_{t-1}, \ldots, t_{t-m+1}$ if we solve by least squares the equations

$$\begin{bmatrix} x_{10} \\ x_{11} \\ x_{12} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \approx \begin{bmatrix} x_0 & x_{-1} & \cdots & x_{-m+1} \\ x_1 & x_0 & & x_{-m+2} \\ x_2 & x_1 & & \cdot \\ x_3 & x_2 & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \\ \cdot & \cdot & & \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \end{bmatrix} \tag{7.4}$$

The matrix in (7.4) may be continued downward for as far as one has data. In an application, the range of t in (7.4) would be over past values of t. Then, after solving the equations for the filter $\mathbf{f}$ it would be hoped that the character of the time series was such that $\mathbf{f}$ could be used to predict future values of the time series which had not gone into the equation defining $\mathbf{f}$.

If the matrix of (7.4) is very much higher than it is wide, it may be desirable to treat the end effects differently. If one uses instead

$$\begin{bmatrix} x_{10} \\ x_{11} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \approx \begin{bmatrix} x_0 & 0 & 0 \\ x_1 & x_0 & 0 \\ x_2 & x_1 & \\ x_3 & x_2 & \\ \vdots & \vdots & \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} \tag{7.5}$$

one finds that the least-squares normal equation has a Toeplitz matrix whereas for (7.4) the matrix in not Toeplitz. As the reader is aware, the Toeplitz matrix has many advantages, both theoretical and computational.

Of special interest is the filter which is designed from the equations

$$\begin{bmatrix} x_0 & & \text{zeros} & \\ x_1 & x_0 & & \\ x_2 & x_1 & x_0 & \\ x_3 & x_2 & \cdot & \ddots \\ \vdots & & \cdot & \\ x_n & & \cdot & \\ & x_n & & \\ & & \ddots & \\ \text{zeros} & & & x_n \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ \\ \cdot \\ \cdot \\ \cdot \\ \\ 0 \end{bmatrix} \tag{7.6}$$

Such a filter is called the *prediction error filter for unit span* because the a_k operate on $(x_{t-1}, x_{t-2}, \ldots)$ attempting to cancel x_t. Thus, the a_k on the $(x_{t-1}, x_{t-2}, \ldots)$ gives the negative of a best prediction of x_t, based on $(x_{t-1}, x_{t-2}, \ldots)$. The normal equations implied by (7.6) are the square set

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots \\ r_1 & r_0 & r_1 & \\ r_2 & r_1 & r_0 & \\ \vdots & & & \\ \cdot & & & \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7.7}$$

It may be noted that the calculation of a prediction error filter depends only on the autocorrelation of the time series and not on the time series itself. As we have seen (from (3.9)), the solutions to these equations are coefficients of a minimum-phase polynomial.

Solutions to Toeplitz equations when the right-hand side takes the more arbitrary form (7.3) are not generally minimum-phase, but the Levinson recursion may be generalized to make the calculation speedy. This is done in Section 7.5 on the multichannel Levinson recursion.

EXERCISES:

1 Find a three-term zero delay inverse to the wavelet (1,2). Compare the error to the error of (2,1). Compare the waveform. An extensive discussion of the error in least-squares inverse filters is given in Reference 26. One conclusion is that the sum of the squared errors goes to zero as the filter length becomes infinite in two situations:

(a) Zero delay inverse if and only if the wavelet being inverted is minimum-phase.

(b) If the wavelet being inverted in not minimum-phase, the error goes to zero only if the output is delayed, that it, $d = (\ldots, 0, 0, 1, 0, 0, \ldots)$. Calculate a three-term delayed inverse to (1,2), that is, try $d = (0, 1, 0, 0)$ or $d = (0, 0, 1, 0)$.

2 A pressure sensor in a deep well records upgoing seismic waves and, at some time t_0 later, identical downgoing waves of opposite sign. Determine delayed and non-delayed least-squares filters of length m to eliminate the double pulse. (You should be able to guess the solution to large matrices of this type. Try filters of the form $f_k = \alpha + \beta k$ where α and β are scalars.) What is the error as a function of the filter length?

3 Let $b_t = (\ldots, 1, 1, -2, 1, 1, -2, \ldots)$. Find by least squares the best one-term filter which predicts b_t, using only b_{t-1}. Find the best two-term filter using b_{t-1} and b_{t-2}. Likewise find the best three-term filter. What is the error as a function of time in each case?

7.2 BURG SPECTRAL ESTIMATION

The uncertainty principle says that if a time function contains most of its energy in the time-span Δt, then its Fourier transform contains most of its energy in a bandwidth $\Delta f \geq 1/\Delta t$. This is not the same as saying that if we have a sample of a stationary time series of length Δt, the best frequency resolution we can hope to attain will be $\Delta f = 1/\Delta t$. The difference lies in the difference between assuming a function is zero outside the interval Δt in which it is given and in assuming that it continues "in a sensible way" outside the given interval. If the data sample can be continued "in a sensible way" some distance beyond the interval in which it is given, then the frequency resolution Δf may be considerably smaller than $1/\Delta t$. A finer resolution depends upon the predictability of the data off the ends of the sample. If one has a segment of a stationary series which is short compared to the autocorrelation of the stationary series, then the spectral estimation procedure of John P. Burg will be radically better than any truncated Fourier transform method. This comes about in physical problems when one is dealing with resonances which have decay times that are long compared to the observation time or when one is looking at a function of space where each point in space represents another instrument.

If a spectrum $R(Z)$ is estimated by $\bar{X}(1/Z)X(Z)$ where $X(Z)$ is a polynomial made up from $N+1$ known data points, then the coefficients of $R(Z)$ are computed by

$$r_k \quad = \quad \sum_{j=0}^{N-k} \bar{x}_{j+k} x_j \tag{7.8}$$

Notice that r_0 is calculated from $N+1$ terms, r_1 from N terms, etc. If N is not large enough, this will have an undesirable biasing effect. The biasing is removed if the r_k are computed instead by the formula

$$r_k \quad = \quad \frac{1}{N-k+1} \sum_{j=0}^{N-k} \bar{x}_{j+k} x_j \tag{7.9}$$

The trouble with using (7.9) is that data samples can easily be found for which r_k will not be a valid autocorrelation function. For example, the spectrum will not be positive at all frequencies, the solution to Toeplitz equations may blow up, etc.

Burg's approach avoids the end-effect problems of (7.8) and the possibility of impossible results from (7.9). Instead of estimating the autocorrelation r_k directly from the data he estimates a minimum-phase prediction-error filter directly from the data. The output of a prediction-error filter has a white spectrum. (If it did not, then the color could be used to improve prediction.) Since the spectrum of the output is the spectrum of the input times the spectrum of the filter, the spectrum of the input may be estimated as the inverse of the spectrum of the prediction-error filter. As we have seen, narrow spectral peaks are far more easily represented by a denominator than by a numerator.

Let the given segment of data be denoted by $x_0, x_1, \ldots, x_n$. Then a two-term prediction-error filter $(1, a)$ of the time series x_t is given by the choice of a which minimizes

$$E(a) \quad = \quad \sum_{t=1}^{N} |x_t + a x_{t-1}|^2 \tag{7.10}$$

Unfortunately, consideration of a few examples shows that there exist time series [like (1,2)] for which $|a|$ may turn out to be greater than unity. This is unacceptable because the prediction-error filter is not minimum-phase, the spectrum is not positive, etc. Recall that a prediction-error filter defined in the previous section depends only on the autocorrelation of the data and not the data per se. This means that the same filter is computed from both a time series and from the (complex-conjugate) time-reversed time series. This suggests that the error of forward prediction (7.10) be augmented by the error of backward prediction. That is

$$E(a) \quad = \quad \sum_{t=1}^{N} |x_t + a x_{t-1}|^2 + |\bar{x}_{t-1} + a \bar{x}_t|^2 \tag{7.11}$$

We will later establish that the minimization of (7.11) always leads to an $|a|$ less than unity. The power spectral estimate associated with this value of a is $R = 1/[(1+\bar{a}/Z)(1+aZ)]$. The value of Δf may be very small if a turns out very close to the unit circle.

A natural extension of (7.11) to filters with more terms would seem to be to minimize

$$E(a_1, a_2) \quad = \quad \sum_{t=2}^{N} |x_t + a_1 x_{t-1} + a_2 x_{t-2}|^2 + |\bar{x}_{t-2} + a_1 \bar{x}_{t-1} + a_2 \bar{x}_t|^2 \tag{7.12}$$

Unfortunately, Burg discovered time series for which the computed filter $A(Z) = 1 + a_1 Z + a_2 Z^2$ was not minimum-phase. If $A(Z)$ is not minimum-phase, then $R = 1/[\bar{A}(1/Z)A(Z)]$ is not a satisfactory spectral estimate because $R(Z)$ is to be evaluated on the unit circle and $1/A(Z)$ would not be convergent there.

Burg noted that the Levinson recursion always gives minimum-phase filters. In the Levinson recursion a filter of order 3 is built up from one of order 2 by

$$\begin{bmatrix} 1 \\ a_1 \\ a_2 \end{bmatrix} \quad = \quad \begin{bmatrix} 1 \\ a \\ 0 \end{bmatrix} - c \begin{bmatrix} 0 \\ a \\ 1 \end{bmatrix}$$

Thus Burg decided that instead of using least squares to determine a_1 and a_2 as in (7.12), he would take a to be given from (7.11) and then do a least-squares problem to solve for c. This would be done in such a way as to ensure that $|c|$ comes out less than unity, which guarantees that $A(Z) = 1 + a_1 Z + a_2 Z^2$ is minimum-phase. Thus he suggested rewriting (7.12) as

$$\begin{aligned} E(c) \quad = \quad & \sum_{t=2}^{N} \quad |x_t + a x_{t-1} - c(\bar{a} x_{t-1} + x_{t-2})|^2 \\ & \quad + \quad |\bar{x}_{t-2} + a \bar{x}_{t-1} - c(\bar{a}\bar{x}_{t-1} + \bar{x}_t)|^2 \end{aligned} \tag{7.13}$$

Now the error (7.13), which is the sum of the error of forward prediction plus the error of backward prediction, is minimized with respect to variation of c. (In a later chapter we will see fit to call c a reflection coefficient.) The quantity a remains fixed by the minimization of (7.11). Now let us establish that $|c|$ is less than unity. Denote by e_+ the time series $x_t + a x_{t-1}$ which is the error in forward prediction of x_t. Denote by e_- the time series $x_{t-2} + \bar{a} x_{t-1}$ of error on backward prediction. With this, (7.13) becomes

$$\begin{aligned} E \quad &= \quad \sum_t |e_+ - c e_-|^2 + |\bar{e}_- - c \bar{e}_+|^2 \\ &= \quad \sum_t \overline{(e_+ - c e_-)}(e_+ - c e_-) + \overline{(\bar{e}_- - c \bar{e}_+)}(\bar{e}_- - c \bar{e}_+) \end{aligned} \tag{7.14}$$

Setting the derivative with respect to $\bar{c}$ equal to zero

$$\begin{aligned} 0 \quad &= \quad \sum_t \bar{e}_-(e_+ - c e_-) + e_+(\bar{e}_- - c \bar{e}_+) \\ c \quad &= \quad \frac{+\sum_t 2\bar{e}_- e_+}{\sum_t \bar{e}_+ e_+ + \bar{e}_- e_-} \end{aligned} \tag{7.15}$$

(One may note that $\partial E/\partial c = 0$ gives the same result.) That $|c|$ is always less than unity may be seen by noting that the length of the vector $e_+ \pm e_-$ is always positive. In particular

$$\begin{aligned} \sum_t |e_+ \pm e_-|^2 &\geq 0 \\ \sum_t \bar{e}_+ e_+ \pm \bar{e}_+ e_- \pm \bar{e}_- e_+ + \bar{e}_- e_- &\geq 0 \\ \sum_t \bar{e}_+ e_+ + \bar{e}_- e_- &> 2|\bar{e}_- e_+| \\ |c| &\leq 1 \end{aligned} \tag{7.16}$$

If we now redefine e_+ and e_- as

$$e_+ \leftarrow e_+ - c e_- \tag{7.17}$$

$$e_- \leftarrow e_- - \bar{c} e_+ \tag{7.18}$$

we have the forward and backward prediction errors of the three-term filter $(1, a_1', a_2') = (1, a_1 - c\bar{a}_1, -c)$. One can then return to (7.14) and proceed recursively. As the recursion proceeds e_+ and e_- gradually become unpredictable random numbers. We have then found a filter $A(Z)$ which filters $X(Z)$ either forward or backward and the output must be the inverse of the spectrum of the filter.

In later chapters we will discover a wave-propagation interpretation of the Burg algorithm. In a layered medium the parameters c_k have the interpretation of the up- and downgoing waves; and the whole process of calculating a succession of c_k amounts to downward continuing surface seismograms into the earth, determining an earth model c_k as you go.

EXERCISES:

1 Consider the time series with ten points $(1, 1, 1, -1, -1, -1, 1, 1, 1, -1)$. Compute C and A up to cubics in Z. Compare the autocorrelation r_t calculated by Burg's method with $R(Z)$ estimated from the truncated sample and with $R(Z)$ estimated by intuitively extending the data sample in time to plus and minus infinity.

2 Modify the program of Figure **??** to compute and include the scale factor V which belongs in the spectrum.

7.3 ADAPTIVE FILTERS

An adaptive filter is one which changes with time to accommodate itself to changes in the time series being filtered. For example, suppose one were predicting one point ahead in a time series. One could take a lot of past data to design the filter; then one could apply the filter to present incoming data to predict future incoming data.

Below is a computer program to do the Burg algorithm. The program follows the notation of the text. The data X is a vector of dimension given to be LX. Choice of $LC \leq LX$ is a compromise between high resolution and high scatter. The density of points on the frequency axis, which is controlled by $N2048 \gg LX$, is chosen for plotting convenience and should be great enough to resolve narrow spectral lines.

```
      SUBROUTINE BURGC (LX,X,EP,EM,LC,C,A,N2048,S)
C GIVEN A TIME SERIES X(1...LX) GET ITS LOG SPECTRUM S(1...N2048)
      DIMENSION X(LX),EP(LX),EM(LX),C(LC),A(LC),S(N2048)
      COMPLEX X,EP,EM,C,A,S,TOP,BOT,EPI,CONJG,CLOG
      DO 10 I=1,N2048
10    S(I)=0.
      A(1)=1.
      DO 20 I=1,LX
      EM(I)=X(I)
20    EP(I)=X(I)
      DO 60 J=2,LC
      TOP=0.
      BOT=0.
      DO 30 I=J,LX
      BOT=BOT+EP(I)*CONJG(EP(I))+EM(I-J+1)*CONJG(EM(I-J+1))
30    TOP=TOP+EP(I)*CONJG(EM(I-J+1))
      C(J)=2*TOP/BOT
      DO 40 I=J,LX
      EPI=EP(I)
      EP(I)=EP(I)-C(J)*EM(I-J+1)
40    EM(I-J+1)=EM(I-J+1)-CONJG(C(J))*EPI
      A(J)=0.
      DO 50 I=1,J
50    S(I)=A(I)-C(J)*CONJG(A(J-I+1))
      DO 60 I=1,J
60    A(I)=S(I)
      CALL FORK(N2048,S,+1.)
      DO 70 I=1,N2048
70    S(I)=-CLOG(S(I))*2.
      RETURN
      END
```

As time goes on it might become desirable to recompute the filter on the basis of new data which have come in. How often should the filter be redesigned? In concept, there is no reason why it should not be recomputed very often, perhaps after each new data point arrives. In practice, this is usually prohibitively expensive. For a filter of length n it requires n multiplies and adds to apply the filter to get one new output point. To recompute the filter with Levinson recursion requires about n^2 multiply-adds. However, it is usually expected that the filter need only be changed by a very small amount when a new data point arrives. For that reason we will give the Widrow adaptive-filter algorithm which modifies the filter by means of only n arithmetic operations. Thus, a new filter is computed after each data point comes in.

For definiteness, consider a two-term prediction situation where e_t is the error in predicting a time series x_t from two of its past values

$$e_t \quad = \quad x_t - bx_{t-1} - cx_{t-2} \tag{7.19}$$

The sum squared error in the prediction is

$$E \quad = \quad \sum_t {e_t}^2 \quad = \quad \sum_t (x_t - bx_{t-1} - cx_{t-2})^2 \tag{7.20}$$

If the parameter b has been chosen correctly, one should find that $\partial E/\partial b = 0$. However, if the nature of the time series x_t is changing with time, $\partial E/\partial b$ may depart from zero when new data are included in the sum in (7.20). Since E is a positive quadratic function of b, if $\partial E/\partial b$ has become positive, then b should be reduced. If $\partial E/\partial b$ has become negative, then b should be augmented. See Figure 7.1.

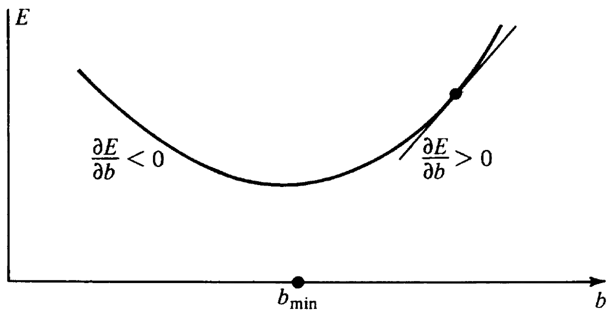

Figure 7.1: The sign of the partial derivative tells whether $b > b_{\text{min}}$ or $b < b_{\text{min}}$. $\boxed{\text{c7-7-2}}$ [NR]

From (7.20) we have

$$\frac{\partial E}{\partial b} \quad = \quad \sum_{i=-\infty}^{t} 2e_i x_{i-1} \tag{7.21}$$

The change in $\partial E/\partial b$ from the addition of the data point x_t is just $-2e_t x_{t-1}$; thus, we are motivated to modify b and c in the following way

$$\begin{bmatrix} b \\ c \end{bmatrix} \leftarrow \begin{bmatrix} b \\ c \end{bmatrix} + ke_t \begin{bmatrix} x_{t-1} \\ x_{t-2} \end{bmatrix} \tag{7.22}$$

Here the number k scales the amount of the readjustment which we are willing to make to b and c in one time step. If k is chosen very small, the adjustment will take place very slowly. If k is chosen too large, the adjustment will overshoot the minimum; however one may hope that it will bounce back, perhaps again overshooting at the next step. The choice of k is dictated in part by the nature of the time series x_t under study.

There are many variations on these same ideas. For example, we could use the L_1 norm and minimize something like

$$E(c) \quad = \quad \sum_t |cx_t - y_t| \tag{7.23}$$

The resulting adaptation would be

$$c \leftarrow c - k x_t \ \mathrm{sgn}\ (c x_t - y_t) \tag{7.24}$$

Equation (7.23) is of course the weighted median. An even more robust procedure is the uniformly weighted median

$$E(c) \quad = \quad \sum_t \left| c - \frac{y_t}{x_t} \right| \tag{7.25}$$

which leads to the adaptation

$$c \leftarrow c - k \ \mathrm{sgn} \left(c - \frac{y_t}{x_t} \right) \tag{7.26}$$

which is identical to

$$c \leftarrow c - k \ \mathrm{sgn}\ (x_t) \ \mathrm{sgn}\ (c x_t - y_t) \tag{7.27}$$

The examples (7.23) and (7.25) could be extended, in a manner like the Burg algorithm, to stationary series. Like (7.25) we could minimize

$$E \quad = \quad \sum_t \left| c - \frac{y_t}{x_t} \right| + \left| c - \frac{x_t}{y_t} \right| \tag{7.28}$$

This leads to a choice of c within the proper bounds because

$$-1 \leq \underset{(\text{all } t)}{\text{median}} \left(0, \tfrac{y_t}{x_t}, \tfrac{x_t}{y_t} \right) \leq +1$$

EXERCISES:

1 If x_t has physical dimensions of volts, what should be the physical dimensions for k? If x_t has an rms value of 100 V and Δt, the sampling interval, is 1 ms, what numerical value of k will allow the Widrow filter to adapt to new conditions in about a second?

2 Consider the time series $x_t = (\ldots, 1, 1, 1, 1, -4, 1, 1, 1, 1, -4, 1, 1, 1, 1, -4, \ldots)$. Consider self-prediction of the form $x_{t+1} = c x_t$. What are the results of least-squares prediction? What are the results of L_1 norm prediction of data weighted and uniformly weighted types?

7.4 DESIGN OF MULTICHANNEL FILTERS

Multichannel filters are frequently useful. For example, with a vector-prediction filter one might wish to predict a time series, using its past and the past of a group of other series. With a matrix-prediction filter one could predict a group of series, using the past of the whole group. If the series are related, the group prediction should be better than self-prediction of individual channels. For definiteness, let us take two time series x_t and y_t and suppose we are to find a vector filter which converts them into a third series d_t. If d_t is x_{t+1}, this is a unit time-span prediction for filter for x_t. If d_t is a vertical seismogram and x_t and y_t are horizontals, then the two-channel filter might be called an extrapolation filter. The set of equations which we wish to solve by least squares takes the form

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \\ \vdots \\ \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & & & & & \\ x_2 & y_2 & x_1 & y_1 & & & \\ x_3 & y_3 & x_2 & y_2 & x_1 & y_1 & \\ x_4 & y_4 & x_3 & y_3 & x_2 & y_2 & \ddots \\ \vdots & & & & & & \\ & & & & & & \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \\ \vdots \\ a_m \\ b_m \end{bmatrix} \tag{7.29}$$

If this set of equations is abbreviated

$$\mathbf{d} \quad \approx \quad \mathbf{Bf} \tag{7.30}$$

then, as we have seen in an earlier chapter, the solution is of the form

$$\mathbf{f} \quad = \quad (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{d} \tag{7.31}$$

We wish to inspect the matrix being inverted, call it $\mathbf{R}$. For a filter with three time lags we get

$$\mathbf{R} \quad = \quad \sum_t \begin{bmatrix} x_t \\ y_t \\ x_{t-1} \\ y_{t-1} \\ x_{t-2} \\ y_{t-2} \end{bmatrix} [x_t \;\; y_t \;\; x_{t-1} \;\; y_{t-1} \;\; x_{t-2} \;\; y_{t-2}] \tag{7.32}$$

If we define

$$\begin{aligned} r_{xx}(i) \quad &= \quad \sum_t x_t x_{t+i} \\ r_{xy}(i) \quad &= \quad \sum_t x_t y_{t+i} \end{aligned}$$

and likewise for $r_{yx}(i)$ and $r_{yy}(i)$ the matrix (7.32) becomes

$$\mathbf{R} \quad = \quad \left[\begin{array}{cc:cc:cc} r_{xx}(0) & r_{xy}(0) & r_{xx}(-1) & r_{xy}(-1) & r_{xx}(-2) & r_{xy}(-2) \\ r_{yx}(0) & r_{yy}(0) & r_{yx}(-1) & r_{yy}(-1) & r_{yx}(-2) & r_{yy}(-2) \\ \hdashline r_{xx}(1) & r_{xy}(1) & r_{xx}(0) & r_{xy}(0) & r_{xx}(-1) & r_{xy}(-1) \\ r_{yx}(1) & r_{yy}(1) & r_{yx}(0) & r_{yy}(0) & r_{yx}(-1) & r_{yy}(-1) \\ \hdashline r_{xx}(2) & r_{xy}(2) & r_{xx}(1) & r_{xy}(1) & r_{xx}(0) & r_{xy}(0) \\ r_{yx}(2) & r_{yy}(2) & r_{yx}(1) & r_{yy}(1) & r_{yx}(0) & r_{yy}(0) \end{array}\right] \tag{7.33}$$

We may take the 6 x 6 matrix of (7.33) and partition it into a 3 x 3 matrix of 2 x 2 submatrices. If we define the submatrix blocks as

$$\mathbf{R}(\tau) \quad = \quad \left[\begin{array}{cc} r_{xx}(\tau) & r_{xy}(\tau) \\ r_{yx}(\tau) & r_{yy}(\tau) \end{array}\right] \quad = \quad \mathbf{R}^T(-\tau) \tag{7.34}$$

then (7.33) in terms of the blocks defined in (7.34) is

$$\mathbf{R} \quad = \quad \left[\begin{array}{lll} R(0) & R(-1) & R(-2) \\ R(1) & R(0) & R(-1) \\ R(2) & R(1) & R(0) \end{array}\right] \tag{7.35}$$

The matrix in (7.35) is called *block Toeplitz* or *multichannel Toeplitz*. As with the ordinary Toeplitz matrix there is a trick method of solution. It will be taken up in the next section.

The reader should note that the matrix **R** does not depend on the desired output **d**. This results in a computational saving when there is more than one possible output. An example would be when it is desired to predict several different series or distances into the future on a given series.

EXERCISES:

1 In the exercises of Chapter 2, we determined $B(Z)$ and $A(Z)$ such that some given power series $C(Z)$ was expressed as $C(Z) = B(Z)/A(Z)$. Write normal equations (do not solve them) for doing this in an approximate way by minimizing

$$\min\,(A,B) \quad = \quad \sum_t (B_t - \sum_t C_{t-\tau} A_\tau)^2$$

where

$$A \quad = \quad (A_0, A_1, A_2) \quad B \quad = \quad (B_0, B_1, B_2)$$

subject to the constraint $A_0 - 1$. (It can be proved that $A(Z)$ comes out minimum-phase by examining the Levinson recursion.)

7.5 LEVINSON RECURSION

The Levinson recursion is a simplified method for solving normal equations. It may be shown to be equivalent to a recurrence relation in orthogonal polynomial theory. The simplification is Levinson's method is possible because the matrix $\mathcal{R}$ has actually only N different elements when a general matrix could have N^2 different elements.

Levinson developed his recursion with single time series in mind (the basic idea was presented in Section 3.3). It is very little extra trouble to do the recursion for multiple time series. Let us begin with the prediction-error normal equation. With multiple time series, unlike single time series, the prediction problem is changed if time is reversed. We may write both the forward and the backward prediction-error normal equations as one equation in the form of (7.36).

Since end effects play an important role, we will show how, when given the solution for 3-term filters, $\mathcal{A}$ and $\mathcal{B}$

$$\begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_{-1} & \mathbf{R}_{-2} \\ \mathbf{R}_1 & \mathbf{R}_0 & \mathbf{R}_{-1} \\ \mathbf{R}_2 & \mathbf{R}_0 & \mathbf{R}_0 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{B}_2 \\ \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_B \end{bmatrix} \tag{7.36}$$

to find the solution $\mathcal{A}'$ and $\mathcal{B}'$ four-term filters to

$$\begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_{-1} & \mathbf{R}_{-2} & \mathbf{R}_{-3} \\ \mathbf{R}_1 & \mathbf{R}_0 & \mathbf{R}_{-1} & \mathbf{R}_{-2} \\ \mathbf{R}_2 & \mathbf{R}_1 & \mathbf{R}_0 & \mathbf{R}_{-1} \\ \mathbf{R}_3 & \mathbf{R}_2 & \mathbf{R}_1 & \mathbf{R}_0 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{B}_3 \\ \mathbf{A}_1 & \mathbf{B}_2 \\ \mathbf{A}_2 & \mathbf{B}_1 \\ \mathbf{A}_3 & \mathbf{I} \end{bmatrix}' = \begin{bmatrix} \mathbf{V}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_B \end{bmatrix}' \tag{7.37}$$

by forming a linear combination of $\mathcal{A}$ and $\mathcal{B}$. This can be done by choosing constant matrices α and β in

$$\begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_{-1} & \mathbf{R}_{-2} & \mathbf{R}_{-3} \\ \mathbf{R}_1 & \mathbf{R}_0 & \mathbf{R}_{-1} & \mathbf{R}_{-2} \\ \mathbf{R}_2 & \mathbf{R}_1 & \mathbf{R}_0 & \mathbf{R}_1 \\ \mathbf{R}_3 & \mathbf{R}_2 & \mathbf{R}_1 & \mathbf{R}_0 \end{bmatrix} \left\{ \begin{bmatrix} \mathbf{I} \\ \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{0} \end{bmatrix} \alpha + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_2 \\ \mathbf{B}_1 \\ \mathbf{I} \end{bmatrix} \beta \right\} = \begin{bmatrix} \mathbf{V}_A \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{E}_A \end{bmatrix} \alpha + \begin{bmatrix} \mathbf{E}_B \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{V}_B \end{bmatrix} \beta \tag{7.38}$$

Make $\mathcal{A}$ by choosing α and β so that the bottom element on the right-hand side of (7.38) vanishes. That is, $\alpha = \mathbf{I}, \beta + -\mathbf{V}_B^{-1}\mathbf{E}_A$. Make $\mathcal{B}$ by choosing α and β so that the top element on the right-hand side vanishes. That is, $\beta = \mathbf{I}, \alpha = -\mathbf{V}_A^{-1}\mathbf{E}_B$.

Of course, one will want to solve more than just the prediction-error problem. We will also want to go from 3 x 3 to 4 x 4 in the solution of the filter problem with arbitrary right-hand side $\mathcal{G}$. This is accomplished by choosing γ in the following construction (7.39) so that $\mathbf{E}_f + \mathbf{V}_{B\gamma} = \mathbf{G}_3$

$$\begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_{-1} & \mathbf{R}_{-2} & \mathbf{R}_{-3} \\ \mathbf{R}_1 & \mathbf{R}_0 & \mathbf{R}_{-1} & \mathbf{R}_{-2} \\ \mathbf{R}_2 & \mathbf{R}_1 & \mathbf{R}_0 & \mathbf{R}_1 \\ \mathbf{R}_3 & \mathbf{R}_2 & \mathbf{R}_1 & \mathbf{R}_0 \end{bmatrix} \left\{ \begin{bmatrix} \mathbf{F}_2 \\ \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_3 \\ \mathbf{B}_2 \\ \mathbf{B}_1 \\ \mathbf{I} \end{bmatrix} \gamma \right\} = \left\{ \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \mathbf{G}_2 \\ \mathbf{E}_f \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{V} \end{bmatrix} \gamma \right\} \tag{7.39}$$

7.6 CONSTRAINED FILTERS

A common geophysical situation is a plane wave (signal) incident on a group of receivers. One expects to see the same waveform at each receiver. However, there is corrupting noise present at each receiver, and the noise may or may not be coherent from one receiver to the next. In fact, we may suppose there is so much noise on each receiver that the signal might not be detectable at all if there were only one receiver. This was the situation facing M.J. Levin when he was trying to detect weak underground nuclear explosions with an array of seismometers. He suggested a multichannel filter with constraints. First suppose that either all the signals arrive at the same time or that, if the times differ, at least they are known so that the data channels may be shifted into alignment. Now the problem is to filter each channel and then add up the channels; the noise should be rejected but the signal shape should be maintained. Let $f_i(j)$ represent the filter weight on the ith channel at the jth lag. For illustration, consider two channels and three lag times. Then Levin's constraints which prevent signal distortion are

$$\begin{aligned} 1 &= f_1(0) + f_2(0) \\ 0 &= f_1(1) + f_2(1) \\ 0 &= f_1(2) + f_2(2) \end{aligned} \tag{7.40}$$

That this does not cause signal distortion follows, since if the same signal $s(Z)$ comes into each channel, the output is merely $s(Z)[f_1(Z) + f_2(Z)]$. But $f_1 + f_2$ is just (1, 0, 0) in this case or a delta function in general. We call the equation set (7.40) constraint equations because there are fewer equations than unknowns. The constraint equations may be written in usual form as

$$\begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ f_1(0) \\ f_2(0) \\ f_1(1) \\ f_2(1) \\ f_1(2) \\ f_2(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.41}$$

which may abbreviate as $\mathbf{Gf} = 0$. If we use the method of least squares to minimize the total energy in the filter output, we will be attempting to suppress both signal and noise. But the constraint equations prevent the suppression of signal; hence only the noise is attenuated. If we let $\mathbf{R}$ denote the spectral matrix of the input data, then the filter $\mathbf{f}$ is determined by solving equations like

$$\begin{bmatrix} \mathbf{R} & \mathbf{G}^T \\ & \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \\ \lambda \end{bmatrix} \quad = \quad \begin{bmatrix} V \\ 0 \\ \vdots \end{bmatrix} \tag{7.42}$$

We have solved equations of this type in preceding sections.

EXERCISES:

1 In one application, where the channel amplifications were not well controlled, the lead terms of the filter were $f_1(0) = 100$ and $f_2(0) = -99$. Although this filter satisfied all that it was designed for, it was deemed inappropriate because the assumption of identical signals on each channel was a reasonable approximation but not exactly true. Can you suggest a more suitable constraint matrix?

2 Consider three seismometers in a row on the surface of the earth. The constraints considered so far have implied all signals arrive at the same time, i.e., vertically incident waves. Define a constraint matrix to pass both the vertically incident wave and the wave which causes $x_1(t) = x_2(t+1) = x_3(t+2)$. What is the sortest filter which can both satisfy the constraints and still have some possibility of rejecting noise?

3 Consider the Levin filter on m channels with filters containing k lags. What is the size of the matrix in (7.42)?