Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.341: DISCRETE-TIME SIGNAL PROCESSING

OpenCourseWare 2006

Lecture 13
The Levinson-Durbin Recursion

In the previous lecture we looked at all-pole signal modeling, linear prediction, and the stochastic inverse-whitening problem. Each scenario was related in concept to the problem of processing a signal $s[n]$ by:

$$s[n] \longrightarrow \boxed{\frac{1}{A}\left[1 - \sum_{k=1}^{p} a_k z^{-k}\right]} \longrightarrow g[n],$$

such that

$$e[n] \equiv g[n] - \delta[n]$$

was minimized in some sense. In the deterministic case, we chose to minimize $\varepsilon = \sum_n e^2[n]$, and in the stochastic case, we chose to minimize $\varepsilon = E\left[e^2[n]\right]$. The causal solution for the parameters a_k was found in each case to be of the form

$$\begin{bmatrix} \phi_s[0] & \phi_s[1] & \cdots & \phi_s[p-1] \\ \phi_s[1] & \phi_s[0] & \cdots & \phi_s[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ \phi_s[p-1] & \phi_s[p-2] & \cdots & \phi_s[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \phi_s[1] \\ \phi_s[2] \\ \vdots \\ \phi_s[p] \end{bmatrix}, \tag{1}$$

or equivalently,

$$T_p \alpha_p = r_p. \tag{2}$$

In this lecture, we address the issue of how exactly to solve for the parameters a_k (or equivalently for the vector α_p). The fact that α_p can be found simply by calculating $\alpha_p = T_p^{-1} r_p$ is worth a mention, but the focus of this lecture is on an elegant technique for finding α_p which is generally less computationally expensive than taking a matrix inversion and which is also recursive in the filter order p. This method for finding α_p is called the *Levinson-Durbin* recursion. In addition to being recursive in p, we'll see that certain intermediate results from the recursion give coefficients for a lattice implementation of an LTI filter $h[n]$ which implements our signal model as

$$H(z) = S'(z) = \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}}.$$

We'll also see that the Levinson-Durbin recursion prescribes a similar implementation for the inverse filter $1/H(z)$.

Let's first prove the Levinson-Durbin recursion. The basic idea of the recursion is to find the solution α_{p+1} for the $(p+1)$st order case from the solution α_p for the pth order case. We'll see that this hinges on the fact that T_p is a $p \times p$ Toeplitz matrix, that is, that T_p is symmetric, and that all entries along a given diagonal are equal. This property of T_p is a result of its definition:

$$(T_p)_{ij} \equiv \phi_s[i-j].$$

To begin the proof, consider what happens in the $p=1$ case. Equation 2 becomes for $p=1$

$$T_1 \alpha_1 = r_1,$$

or

$$\phi_s[0] a_1 = \phi_s[1],$$

giving the solution

$$a_1 = \frac{\phi_s[1]}{\phi_s[0]}. \tag{3}$$

The recursion will now be developed by evaluating Equation 2 for order $p+1$:

$$T_{p+1} \alpha_{p+1} = r_{p+1}.$$

Because we will be dealing with a solution vector α_ℓ for multiple orders ℓ, we will adopt the notation $a_k^{(\ell)}$ to refer to the parameter a_k for the ℓth-order model. (Note that $a_k^{(\ell)}$ is generally not equal to $a_k^{(m)}$.) Equation 2 is therefore represented in matrix form as

$$\begin{bmatrix} \phi_s[0] & \phi_s[1] & \cdots & \phi_s[p-1] & \phi_s[p] \\ \phi_s[1] & \phi_s[0] & \cdots & \phi_s[p-2] & \phi_s[p-1] \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi_s[p-1] & \phi_s[p-2] & \cdots & \phi_s[0] & \phi_s[1] \\ \phi_s[p] & \phi_s[p-1] & \cdots & \phi_s[1] & \phi_s[0] \end{bmatrix} \begin{bmatrix} a_1^{(p+1)} \\ a_2^{(p+1)} \\ \vdots \\ a_p^{(p+1)} \\ a_{p+1}^{(p+1)} \end{bmatrix} = \begin{bmatrix} \phi_s[1] \\ \phi_s[2] \\ \vdots \\ \phi_s[p] \\ \phi_s[p+1] \end{bmatrix}.$$

The matrix T_{p+1} relates to T_p as

$$T_{p+1} = \left[\begin{array}{cccc|c} & & & & \phi_s[p] \\ & & & & \phi_s[p-1] \\ & T_p & & & \vdots \\ & & & & \phi_s[1] \\ \hline \phi_s[p] & \phi_s[p-1] & \cdots & \phi_s[1] & \phi_s[0] \end{array}\right],$$

and the vector r_{p+1} relates to r_p as

$$r_{p+1} = \left[\begin{array}{c} r_p \\ \hline \phi_s[p+1] \end{array}\right].$$

Defining a new vector ρ_p as r_p upside-down, or

$$\rho_p = \begin{bmatrix} \phi_s[p] \\ \phi_s[p-1] \\ \vdots \\ \phi_s[1] \end{bmatrix},$$

gives a more compact representation of T_{p+1} in terms of T_p:

$$T_{p+1} = \left[\begin{array}{c|c} T_p & \rho_p \\ \hline (\rho_p)^T & \phi_s[0] \end{array}\right].$$

Let's now represent the $(p+1)$st-order parameter vector α_{p+1} in terms of the pth order vector α_p, a correction term k_{p+1} and a correction vector ε_p as

$$\boxed{\alpha_{p+1} = \left[\begin{array}{c} \alpha_p \\ \hline 0 \end{array}\right] + \left[\begin{array}{c} \varepsilon_p \\ \hline k_{p+1} \end{array}\right]}. \tag{4}$$

Equation 2 for order $p+1$ is therefore represented in terms of the pth order equation as

$$\left[\begin{array}{c|c} T_p & \rho_p \\ \hline (\rho_p)^T & \phi_s[0] \end{array}\right] \left\{ \left[\begin{array}{c} \alpha_p \\ \hline 0 \end{array}\right] + \left[\begin{array}{c} \varepsilon_p \\ \hline k_{p+1} \end{array}\right] \right\} = \left[\begin{array}{c} r_p \\ \hline \phi_s[p+1] \end{array}\right],$$

which, sorting the equations out, implies

$$T_p \alpha_p + T_p \varepsilon_p + \rho_p k_{p+1} = r_p \tag{5}$$

and

$$(\rho_p)^T \alpha_p + (\rho_p)^T \varepsilon_p + \phi_s[0] k_{p+1} = \phi_s[p+1]. \tag{6}$$

Substituting Equation 2 for order $p = 1$ into Equation 5 then gives

$$T_p \varepsilon_p k_{p+1}^{-1} = -\rho_p. \tag{7}$$

Note now that since T_p is Toeplitz, the matrix realization of the causal Yule-Walker equations for order p (Equation 1) implies also

$$\begin{bmatrix} \phi_s[0] & \phi_s[1] & \cdots & \phi_s[p-1] \\ \phi_s[1] & \phi_s[0] & \cdots & \phi_s[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ \phi_s[p-1] & \phi_s[p-2] & \cdots & \phi_s[0] \end{bmatrix} \begin{bmatrix} a_p^{(p)} \\ a_{p-1}^{(p)} \\ \vdots \\ a_1^{(p)} \end{bmatrix} = \begin{bmatrix} \phi_s[p] \\ \phi_s[p-1] \\ \vdots \\ \phi_s[1] \end{bmatrix},$$

or

$$T_p \beta_p = \rho_p,$$

where β_p is α_p upside-down. This then implies, after substituting in Equation 5,

$$\boxed{\varepsilon_p = -k_{p+1}\beta_p}. \tag{8}$$

We'll now use this and Equation 6 to find k_{p+1} and thus have all the pieces needed to complete the recursion.

Pre-multiplying both sides of Equation 8 by $(\rho_p)^T$ gives the scalar relation

$$(\rho_p)^T \varepsilon_p = -(\rho_p)^T \beta_p k_{p+1}.$$

Note, however, that the scalar relation $(\rho_p)^T \beta_p = (r_p)^T \alpha_p$ also holds, since r_p is a re-ordered version of ρ_p in the same way that α_p is a re-ordered version of β_p. This allows us to re-write the above equation as

$$(\rho_p)^T \varepsilon_p = -(r_p)^T \alpha_p k_{p+1},$$

which implies from Equation 6

$$(\rho_p)^T \alpha_p - (r_p)^T \alpha_p k_{p+1} + \phi_s[0]k_{p+1} = \phi_s[p+1],$$

or

$$\boxed{k_{p+1} = \frac{\phi_s[p+1] - (\rho_p)^T \alpha_p}{\phi_s[0] - (r_p)^T \alpha_p}}. \tag{9}$$

Equations 4, 8, and 9 therefore define a recursion formula for finding the $(p+1)$st-order model parameters $a_k^{(p+1)}$ in terms of the previously-obtained pth-order solution. Equation 3 gives a closed-form solution for $a_1^{(1)}$, which provides a starting point for the recursion. Since this ends a considerable algebraic detour, we'll now summarize what was concluded.

The Levinson-Durbin recursion

- Problem statement: for order p, solve $\sum_{k=1}^{p} a_k^{(p)} \phi_s[i-k] = \phi_s[i]$
- Definitions:

$$\begin{aligned}
\alpha_p &= \left[a_1^{(p)}, a_2^{(p)}, \ldots, a_p^{(p)}\right]^T \\
\beta_p &= \left[a_p^{(p)}, a_{p-1}^{(p)}, \ldots, a_1^{(p)}\right]^T \\
r_p &= [\phi_s[1], \phi_s[2], \ldots, \phi_s[p]]^T \\
\rho_p &= [\phi_s[p], \phi_s[p-1], \ldots, \phi_s[1]]^T
\end{aligned}$$

- Solution for $p=1$: $a_1^{(1)} = \dfrac{\phi_s[1]}{\phi_s[0]}$
- Recursion:

$$k_{p+1} = \frac{\phi_s[p+1] - (\rho_p)^T \alpha_p}{\phi_s[0] - (r_p)^T \alpha_p}$$

$$\varepsilon_p = -k_{p+1}\beta_p$$

$$\alpha_{p+1} = \left[\frac{\alpha_p}{0}\right] + \left[\frac{\varepsilon_p}{k_{p+1}}\right] = \begin{bmatrix} a_1^{(p)} \\ a_2^{(p)} \\ \vdots \\ a_p^{(p)} \\ 0 \end{bmatrix} - k_{p+1} \begin{bmatrix} a_p^{(p)} \\ a_{p-1}^{(p)} \\ \vdots \\ a_1^{(p)} \\ -1 \end{bmatrix}$$

In parting, note that when these coefficients k_ℓ are used to implement an all-pole lattice filter with reflection coefficients k_ℓ, the lattice filter's response is described by

$$H(z) = S'(z) = \frac{A}{1 - \sum_{m=1}^{p} a_m z^{-m}}.$$

In other words, the impulse response of an all-pole lattice filter with reflection coefficients k_ℓ, $\ell = 1, \ldots, p$, as prescribed by the Levinson-Durbin recursion, is the pth-order model of our original signal $s[n]$. (Likewise, the corresponding all-zero lattice filter implements $1/S'(z)$.) Note further that it is straightforward to determine the stability of an all-pole filter when implemented in lattice form. Specifically, if the coefficients k_ℓ all have magnitude < 1, the all-pole filter is guaranteed to be stable. (It is left as an exercise to gain intuition for why this is so.) This stability criterion is of particular interest in applications where the all-pole filter is implemented with finite-precision coefficients, since determining the stability of a comparable direct-form all-pole implementation requires the generally more-expensive process of determining all of the filter's pole locations. In addition to the computational benefits gained from the Levinson-Durbin recursion, its connection with the lattice structure (and the structure's associated stability metric) enables a wide array of applications, including real-time speech coders and synthesizers.