Digital Speech Processing—Lecture 13

Linear Predictive Coding (LPC)—Introduction

LPC Methods

- LPC methods are the most widely used in speech coding, speech synthesis, speech recognition, speaker recognition and verification and for speech storage
  - LPC methods provide extremely accurate estimates of speech parameters, and does it extremely efficiently
  - basic idea of Linear Prediction: current speech sample can be closely approximated as a linear combination of past samples, i.e.,

\[ s(n) = \sum_{k=1}^{p} a_k s(n-k) \text{ for some value of } p, a_k \text{'s} \]

LPC Methods

- for periodic signals with period \( N_p \), it is obvious that
  \[ s(n) = s(n - N_p) \]
  but that is not what LP is doing; it is estimating \( s(n) \) from the \( p \) \( (p << N_p) \) most recent values of \( s(n) \) by linearly predicting its value
- for LP, the predictor coefficients (the \( a_k \)’s) are determined (computed) by minimizing the sum of squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones

LPC Methods

- LP methods have been used in control and information theory—called methods of system estimation and system identification
  - used extensively in speech under group of names including
    1. covariance method
    2. autocorrelation method
    3. lattice method
    4. inverse filter formulation
    5. spectral estimation formulation
    6. maximum likelihood method
    7. inner product method

Basic Principles of LP

- the time-varying digital filter represents the effects of the glottal pulse shape, the vocal tract IR, and radiation at the lips
- the system is excited by an impulse train for voiced speech, or a random noise sequence for unvoiced speech
- this 'all-pole' model is a natural representation for non-nasal voiced speech—but it also works reasonably well for nasals and unvoiced sounds
LP Basic Equations

- A $p$th order linear predictor is a system of the form
  \[ \hat{e}(n) = \sum_{k=0}^{p} a_k e(n-k) = 0 \]
- The prediction error, $e(n)$, is of the form
  \[ \hat{e}(n) = \sum_{k=0}^{p} a_k e(n-k) + \sum_{k=0}^{p} e(n-k) \]  
- The prediction error is the output of a system with transfer function
  \[ H(z) = \frac{1}{A(z)} \]  
- If the speech signal obeys the production model exactly, and if
  \[ e(n) = 0 \]  
- $A(z)$ is an inverse filter for $H(z)$, i.e.,
  \[ H(z) = \frac{1}{A(z)} \]  

Solution for $\{\alpha_k\}$

- Short-time average prediction squared-error is defined as
  \[ E_{\hat{e}} = \sum_m E_{\hat{e}}(m) = \sum_m (s_{\hat{n}}(m) - \hat{e}(m))^2 \]
- Select segment of speech $s_{\hat{n}}(m) = s(m + \hat{n})$ in the vicinity of sample $\hat{n}$
- The key issue to resolve is the range of $m$ for summation
  (to be discussed later)

Solution for $\{\alpha_k\}$

- Defining
  \[ \phi(i,k) = \sum_m s_{\hat{n}}(m) s_{\hat{n}}(m-k) \]
- We get
  \[ \sum_{i=1}^{2p} \alpha_i \phi(i,k) = \phi(i,0), \quad i = 1,2,...,p \]
- Leading to a set of $p$ equations in $p$ unknowns that can be solved in an efficient manner for the $\{\alpha_k\}$

Solution for $\{\alpha_k\}$

- Minimum mean-squared prediction error has the form
  \[ E_{\hat{e}} = \sum_m \sum_{k=0}^{p} a_k \sum_{i=1}^{p} \phi(i,k) s_{\hat{n}}(m-k) \]
- Which can be written in the form
  \[ E_{\hat{e}} = \phi_i(0,0) \sum_{k=0}^{p} a_k \phi_i(0,k) \]

Solution for $\{\alpha_k\}$

- Need to determine $\{\alpha_k\}$ directly from speech such that they give good estimates of the time-varying spectrum
- Need to estimate $\{\alpha_k\}$ from short segments of speech
- Need to minimize mean-squared prediction error over short segments of speech
- Resulting $\{\alpha_k\}$ assumed to be the actual $\{\alpha_k\}$ in the speech production model
- Intend to show that all of this can be done efficiently, reliably, and accurately for speech
Autocorrelation Method

- Assume $s_j(m)$ exists for $0 \leq m \leq L - 1$ and is exactly zero everywhere else (i.e., window of length $L$ samples)
  \[ s_j(m) = s(m + n)w(m), \quad 0 \leq m \leq L - 1 \]
- Where $w(m)$ is a finite length window of length $L$ samples

The "Autocorrelation Method"

- For calculation of $s_j(m)$ since $s_j(m) = 0$ outside the range $0 \leq m \leq L - 1$, then
  \[ s_j(k) = \sum_{m=0}^{L-1} s_j(m)w(m-k), \quad 1 \leq j \leq p, 0 \leq k \leq p \]
- Which is equivalent to the form
  \[ s_j(k) = \sum_{m=0}^{L-1} s_j(m) \delta (m-k), \quad 1 \leq j \leq p, 0 \leq k \leq p \]
- There are $L - 1$ non-zero terms in the computation of $s_j(k)$ for each value of $j$ and $k$ can easily show that
  \[ s_j(k) = R_j(\delta (m-k)), \quad 1 \leq j \leq p, 0 \leq k \leq p \]
- Where $R_j(\delta (m-k))$ is the short-time autocorrelation of $s_j(m)$ evaluated at $-k$ where
  \[ R_j(k) = \sum_{m=0}^{L-1} s_j(m)s_j(m + k) \]

Large errors at ends of window
### Autocorrelation Method

- Since $R_k(k)$ is even, then 
  \[ \phi_k(i, k) = R_k(i - k) \quad 1 \leq i \leq p, 0 \leq k \leq p \]
- Thus the basic equation becomes 
  \[ \sum_{m=0}^{L-1} a_m \phi_m(0, k) = R_k(0) - \sum_{m=0}^{L-1} a_m R_m(i) \quad 1 \leq i \leq p \]
- With the minimum mean-squared prediction error of the form 
  \[ E_k = \phi_k(0, 0) - \sum_{m=0}^{L-1} a_m \phi_m(0, k) \]

### Covariance Method

- There is a second basic approach to defining the speech segment $s_k(m)$ and the limits on the sums, namely **fix the interval** over which the mean-squared error is computed, giving 
  \[(\text{Assumption } \#2): \]
  \[ E_k = \sum_{m=0}^{L-1} \phi_k(0, m) = \sum_{m=0}^{L-1} s_k(m) - \sum_{m=0}^{L-1} a_k s_k(m - k) \]
  \[ \phi_k(i, k) = \sum_{m=0}^{L-1} s_k(m - k) s_k(m), \quad 1 \leq i \leq p, 0 \leq k \leq p \]

### Covariance Method

- Changing the summation index gives 
  \[ \phi_k(i, k) = \sum_{m=0}^{L-1} s_k(m - l) s_k(m - k), \quad 1 \leq i \leq p, 0 \leq k \leq p \]
  \[ \phi_k(i, k) = \sum_{m=0}^{L-1} s_k(m - l) s_k(m - k), \quad 1 \leq i \leq p, 0 \leq k \leq p \]
  \[ \text{key difference from Autocorrelation Method is that limits of summation include terms before m = 0} \Rightarrow \text{window extends p samples backwards from a(0) to a(l - 1)} \]

### Covariance Method

- Since we are extending window backwards, don't need to taper it using a HWP since there is **no transition at window edges**

### Covariance Method

- Cannot use autocorrelation formulation => this is a true cross correlation
  - Need to solve set of equations of the form 
  \[ \sum_{m=0}^{L-1} a_m \phi_m(i, k) - \phi_m(0, 0), \quad i = 1, 2, ..., p \]
  \[ E_k = \phi_k(0, 0) - \sum_{m=0}^{L-1} a_m \phi_m(0, k) \]
  \[ \begin{bmatrix} \phi_1(1) & \phi_1(2) & \cdots & \phi_1(p) \\ \phi_2(1) & \phi_2(2) & \cdots & \phi_2(p) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_p(1) & \phi_p(2) & \cdots & \phi_p(p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \phi_1(0) \\ \phi_2(0) \\ \vdots \\ \phi_p(0) \end{bmatrix} \]
  \[ a = \mathbf{v} \] or \[ a = \mathbf{f}^T \mathbf{v} \]
Covariance Method

- we have \( \phi \{ l, k \} = \phi \{ k, i \} \) \( \Rightarrow \) symmetric but not Toeplitz matrix
  - whose diagonal elements are related as
    \[
    \phi \{ l + 1, k + 1 \} = \phi \{ l, k \} + s_l (-1) \phi \{ l, k - 1 \} - s_{l-1} \phi \{ L - 1 - l, L - 1 - k \} \\
    \phi \{ 2, 2 \} = \phi \{ l, i \} + s_l (-2) \phi \{ -(2-l), -2 \phi \{ L - 2 \} \\
    \]
- all terms \( \phi \{ l, k \} \) have a fixed number of terms contributing to the computed values (L terms)
  - \( \phi \{ l, k \} \) is a covariance matrix \( \Rightarrow \) specialized solution for \( \{ u_a \} \) called the Covariance Method

Summary of LP

- use \( p \) order linear predictor to predict \( s(n) \) from \( p \) previous samples
  - minimize mean-squared error, \( E_\alpha \), over analysis window of duration \( L \) samples
    - solution for optimum predictor coefficients, \( \{ u_a \} \), is based on solving a matrix equation
  - two solutions have evolved
    - autocorrelation method \( \Rightarrow \) signal is windowed by a tapering window in order to minimize discontinuities at beginning (predicting speech from zero-valued samples) and end (predicting zero-valued samples from speech samples) of the interval; the resulting autocorrelation matrix is Toeplitz and can be solved using standard matrix solvers
    - covariance method \( \Rightarrow \) the signal is extended by \( p \) samples outside the normal range of \( 0 \leq m \leq L - 1 \) to include \( p \) samples occurring prior to \( m = 0 \); this eliminates large errors in computing the signal from values prior to \( m = 0 \) (they are available) and eliminates the need for a tapering window; resulting matrix of correlations is symmetric but not Toeplitz
  - different method of solution with somewhat different set of optimal prediction coefficients, \( \{ u_a \} \)

LPC Summary

1. Speech Production Model:
   \[
   s(n) - \sum_{k=0}^{L-1} a_k s(n-k) + G(n)
   \]
   \[
   H(z) = \frac{S(z)}{G(z)} = \frac{1}{1 - \sum_{k=0}^{L-1} a_k z^{-k}}
   \]

2. Linear Prediction Model:
   \[
   \hat{s}(\delta) - \sum_{i=0}^{L-1} s_{\alpha}(\delta - k)
   \]
   \[
   \hat{P}(z) = \frac{S(z)}{S(z)} - 1 - \sum_{k=0}^{L-1} a_k z^{-k}
   \]

3. LPC Minimization:
   \[
   E_\alpha = \frac{1}{p} \sum_{i=0}^{p-1} E_i(\alpha) \Rightarrow \arg \min_{\alpha} E_\alpha
   \]
   \[
   E_i(\alpha) = \sum_{m=0}^{L-1} s(m) - \sum_{k=0}^{p-1} \alpha_k s(m-k)
   \]

4. Autocorrelation Method:
   \[
   s_{\alpha}(m) = s(m) + Hw(m), \quad 0 \leq m \leq L - 1
   \]
   \[
   s_0(m) - s_0(m-r) - \sum_{k=0}^{L-1} a_k s_0(m-k), \quad 0 \leq m \leq L - 1 + p
   \]
   \( \Rightarrow \) large errors for \( 0 \leq m \leq p - 1 \) and for \( L \leq m \leq L + p \)
   \[
   E_i = \sum_{m=0}^{L-1} s_{\alpha}(m)
   \]
   \[
   \phi \{ l, k \} = R_i(l-k) - \sum_{m=0}^{L-1} s_{\alpha}(m) s_{\alpha}(m+l-k) - R_i[l+k]
   \]
   \[
   \phi \{ 0, 0 \} = R_0(0) - \sum_{i=0}^{p} R_i(k)
   \]
LPC Summary

5. Covariance Method:
   - For interval for error signal
     \[ E_z = \sum_{k=0}^{\infty} r_z(m) - \sum_{l=0}^{\infty} a_l r_z(m-l) \]
   - Need signal for \( \alpha(0 - p) \) to \( \alpha(1 - L - 1) \)
   - \( E_z = 0, L = 1 \)
   - \[ E_z = \sum_{k=0}^{\infty} \phi(k) - \sum_{k=0}^{\infty} \phi(k) \phi(k) \]
   - \( \phi(k) \) symmetric matrix:
     \[ \begin{bmatrix}
     \phi(0,0) & \phi(0,1) & \ldots & \phi(0,p) \\
     \ldots & \ldots & \ldots & \ldots \\
     \phi(p,0) & \phi(p,1) & \ldots & \phi(p,p) \\
     \end{bmatrix} \]

Computation of Model Gain

- It is reasonable to expect the model gain, \( G \), to be determined by matching the signal energy with the energy of the linearly predicted samples.
- From the basic model equations, we have:
  \[ G = \frac{u(n)}{a(n)} \sum_{k=0}^{\infty} \phi(k) \phi(k) \]
  \( \Rightarrow \) model
- Whereas for the prediction error we have:
  \[ u(n) = \sum_{k=0}^{\infty} \phi(k) \phi(k) \]
- When \( \phi(k) = 0 \), a perfect match to model.
- Since it is impossible to guarantee that \( \phi(k) = 0 \), this simple matching property for determining \( G \), instead use energy matching criterion (energy in error signal)(energy in excitation)
  \[ G = \frac{\sum_{k=0}^{\infty} u^2(k)}{\sum_{k=0}^{\infty} \phi(k) \phi(k)} \]

Gain Assumptions

- Assumptions about excitation to solve for \( G \):
  - Voiced speech: \( u(n) = \alpha(n) \) \( \Rightarrow \) \( \alpha(n) \) order of a single pitch period; predictor order, \( p \), large enough to model glottal pulse shape, vocal tract IR, and radiation
  - Unvoiced speech: \( u(n) = \alpha(n) \) zero mean, unity variance, stationary white noise process

Solution for Gain (Voiced)

- Since \( \tilde{R}(m) \) and \( R_z(m) \) have the identical form, for \( c \) constant to be determined.
- If total energies of the signal \( \tilde{R}(0) \) and the impulse response \( R_z(0) \) must be equal, the constant \( c \) must be 1, and we obtain the relation
  \[ \sum_{k=0}^{\infty} \phi(k) \phi(k) = E_z \]
- Since \( \tilde{R}(m) = R_z(m) \), \( m = p \), and the energy of the impulse response \( \alpha(n) \) to first \( p+1 \) coefficients of the autocorrelation of the impulse response of the model are identical to the first \( p+1 \) coefficients of the autocorrelation function of the speech signal. This condition called the autocorrelation matching property of the autocorrelation method.

Solution for Gain (Unvoiced)

- For unvoiced speech the input is white noise with zero mean and unity variance, i.e.,
  \[ \frac{E[u(n)u(n-m)]}{E[u(n)]} = \delta(m) \]
- If we excite the system with input \( Gu(n) \) and call the output \( \tilde{g}(n) \) then
  \[ \tilde{g}(n) = \sum_{k=0}^{\infty} \phi(n-k) Gu(n) \]
- Since the autocorrelation function for the output is the convolution of the autocorrelation function of the impulse response with the autocorrelation function of the white noise input, then
  \[ E[\tilde{g}(n)\tilde{g}(n-m)] = R\tilde{g}(m) \]
  \[ \Rightarrow R\tilde{g}(m) = R(0) \sum_{k=0}^{\infty} \phi(k) \phi(k) \]
  \( \Rightarrow \) since \( \phi(k) = 0 \) for \( m > 0 \) because \( u(n) \) is uncorrelated with any signal prior to \( u(n) \)
Solution for Gain (Unvoiced)

- for \( m = 0 \) we get
  \[
  \hat{R}(m) = \sum_{k=1}^{\infty} a_k \tilde{R}(k) + GE[u(n)\tilde{g}(n)]
  \]
  \[
  = \sum_{k=1}^{\infty} a_k \tilde{R}(k) + G^2
  \]
- since \( E[u(n)\tilde{g}(n)] = E[u(n)G\tilde{u}(n) \text{ terms prior to } n] = G^2 \)
- since the energy in the signal must equal the energy in the response to \( \tilde{G}(n) \) we get
  \[
  \hat{R}(m) = R_G(m)
  \]
  \[
  G^2 = R_G(0) = \sum_{k=1}^{\infty} a_k R_G(k) = E_u
  \]

The Resulting LPC Model

- The final LPC model consists of the LPC parameters, \( \{a_k\}, k = 1,2,\ldots,p \), and the gain, \( G \), which together define the system function
  \[
  \hat{R}(z) = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}}
  \]
- with frequency response
  \[
  \hat{H}(e^{j\omega}) = \frac{G}{1- \sum_{k=1}^{p} a_k e^{-j\omega k}}
  \]
- with the gain determined by matching the energy of the model to the short-time energy of the speech signal, i.e.,
  \[
  G^2 = E_u - \sum_{k=1}^{p} (a_k(m))^2 - R_G(0) \sum_{k=1}^{\infty} a_k R_G(k)
  \]

LP Short-Time Spectrum Analysis

- Defined speech segment as:
  \[
  s_n[m] = s[m+\tilde{n}]w[m]
  \]
- The discrete-time Fourier transform of this windowed segment is:
  \[
  S_x(e^{j\omega}) = \sum s[m+\tilde{n}]w[m]e^{-j\omega n}
  \]
- Short-time FT and the LP spectrum are linked via short-time autocorrelation

Frequency Domain Interpretations of Linear Predictive Analysis

LPC Spectrum

LP Short-Time Spectrum Analysis

(a) Voiced speech segment obtained using a Hamming window
(b) Corresponding short-time autocorrelation function used in LP analysis (heavy line shows values used in LP analysis)
(c) Corresponding short-time log magnitude Fourier transform and short-time log magnitude LPC spectrum (FS=16 kHz)
LP Short-Time Spectrum Analysis

(a) Unvoiced speech segment obtained using a Hamming window
(b) Corresponding short-time autocorrelation function used in LP analysis (heavy line shows values used in LP analysis)
(c) Corresponding short-time log magnitude Fourier transform and short-time log magnitude LPC spectrum (Fs = 16 kHz)

Frequency Domain Interpretation of Mean-Squared Prediction Error

The LP spectrum provides a basis for examining the properties of the prediction error (or equivalently the excitation of the VT)

The mean-squared prediction error at sample \( \hat{a} \) is:

\[
E_\epsilon = \sum_{n=0}^{\infty} \epsilon[n]^2
\]

which, by Parseval's Theorem, can be expressed as:

\[
E_\epsilon = \frac{1}{2\pi} \int_{-\pi}^{\pi} |S_\epsilon(e^{j\omega})|^2 \, d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\hat{A}(e^{j\omega})|^2 \, d\omega = G^2
\]

where \( S_\epsilon(e^{j\omega}) \) is the FT of \( \epsilon[n] \) and \( \hat{A}(e^{j\omega}) \) is the corresponding prediction error frequency response

\[
\hat{A}(e^{j\omega}) = 1 - \sum_{l=1}^{\infty} a_l e^{-j\omega l}
\]

LP Interpretation Example1

Much better spectral matches to STFT spectral peaks than to STFT spectral valleys as predicted by spectral interpretation of error minimization.

LP Interpretation Example2

Note small differences in spectral shape between STFT, autocorrelation spectrum and covariance spectrum when using short window duration (L = 51 samples).

Effects of Model Order

The AC function, \( R_s[m] \) of the speech segment, \( s[m] \), and the AC function, \( \hat{R}[m] \), of the impulse response, \( \hat{h}[m] \), corresponding to the system function, \( \hat{H}(z) \), are equal for the first \( (p + 1) \) values. Thus, as \( p \to \infty \), the AC functions are equal for all values and thus:

\[
\lim_{p \to \infty} |\hat{H}(e^{j\omega})|^2 = |S_s(e^{j\omega})|^2
\]

Thus if \( p \) is large enough, the FR of the all-pole model, \( \hat{H}(e^{j\omega}) \), can approximate the signal spectrum with arbitrarily small error.
Effects of Model Order

Plots show Fourier transform of segment and LP spectra for various orders.
- As \( p \) increases, more details of the spectrum are preserved.
- Need to choose a value of \( p \) that represents the spectral effects of the glottal pulse, vocal tract and radiation—nothing else.

Effects of Model Order

Linear Prediction Spectrogram

Speech spectrogram previously defined as:

\[
20 \log |S[k]| = 20 \log \left| \sum_{n} [R + \mu]e^{j2\pi \frac{n}{N} \omega} \right|
\]

for set of times, \( \omega = 2\pi \), and set of frequencies, \( F_k = kF_s/N, \ k = 1,2,...,N/2 \)

where \( \mu \) is the time shift (in samples) between adjacent STFTs, \( T \) is the sampling period, \( F_s = 1/T \) is the sampling frequency, and \( N \) is the size of the discrete Fourier transform used to compute each STFT estimate.

Similarly we can define the LP spectrogram as an image plot of:

\[
20 \log |\hat{R}[k]| = 20 \log \left\{ \sum_{r} \frac{G_r}{|e^{2\pi \frac{r}{N} \omega}-1|} \right\}
\]

where \( G_r \) and \( A(e^{j2\pi \frac{r}{N} \omega}) \) are the gain and prediction error polynomial at analysis time \( r \).
Comparison to Other Spectrum Analysis Methods

Spectra of synthetic vowel /IY/
(a) Narrowband spectrum using 40 msec window
(b) Wideband spectrum using a 10 msec window
(c) Cepstrally smoothed spectrum
(d) LPC spectrum from a 40 msec section using a p=12 order LPC analysis

Selective Linear Prediction

- It is possible to apply LP methods to selected parts of spectrum
  - 0-4 kHz for voiced sounds → use a predictor of order \(p_1\)
  - 4-8 kHz for unvoiced sounds → use a predictor of order \(p_2\)
- The key idea is to map the frequency region \((f_a, f_b)\) linearly to \([0, \pi)\)
or, equivalently, the region \([2\pi f_a, 2\pi f_b]\) maps linearly to \([0, \pi]\) via the transformation
  \[\phi = \frac{\omega - 2\pi f_a}{2\pi f_b - 2\pi f_a}\]
- We must modify the calculation for the autocorrelation to give:
  \[R'(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{i=1}^{\pi} \{e^{j\omega i}\}^*)e^{j\omega j}d\omega\]

Solutions of LPC Equations

Covariance Method (Cholesky Decomposition Method)

LPC Solutions-Covariance Method

- For the covariance method we need to solve the matrix equation
  \[\sum_{k=1}^{\infty} a_k \phi_k(i,k) - \phi_k(i,0) = 0, \quad i = 1, 2, ..., p\]
  \(\phi = \phi, \quad \phi = \phi\) (matrix notation)
- \(\phi\) is a positive definite, symmetric matrix with \((i,j)\) element \(\phi_k(i,j)\),
  and \(\phi\) and \(\phi\) are column vectors with elements \(a_i\) and \(\phi_k(i,k)\)
- The solution of the matrix equation is called the Cholesky decomposition, or square root method
  \[\rho = \Phi \sqrt{V}; \quad V = \text{diagonal matrix}\]
LPC Solutions-Covariance Method

- can readily determine elements of V and D by solving for \((i,j)\) elements of the matrix equation, as follows
  \[ \hat{\phi}(i,j) = \sum_{k=1}^{i-1} V_{ik} \hat{d}_k, \quad 1 \leq j < i \]
  giving
  \[ V_{ij} \hat{d}_j = \hat{\phi}(i,j) - \sum_{k=1}^{j-1} V_{ik} \hat{d}_k, \quad 1 \leq j < i \]
  and for the diagonal elements
  \[ \hat{\phi}(i,i) = \sum_{k=1}^{i-1} V_{ik} \hat{d}_k, \quad i \geq 2 \]
  giving
  \[ d_i = \hat{\phi}(i,i) - \sum_{k=1}^{i-1} V_{ik} \hat{d}_k, \quad i \geq 2 \]
  with
  \[ \hat{d}_i = \hat{\phi}(i,i) \]

Cholesky Decomposition Example

- consider example with \(p = 4\), and matrix elements \(\phi(i,j) = \phi_j\)
  \[
  \begin{bmatrix}
  \phi_1 & \phi_2 & \phi_3 & \phi_4 \\
  \phi_1 & \phi_2 & \phi_3 & \phi_4 \\
  \phi_1 & \phi_2 & \phi_3 & \phi_4 \\
  \phi_1 & \phi_2 & \phi_3 & \phi_4 \\
  \end{bmatrix}
  \]
  \[
  \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1 \\
  \end{bmatrix}
  \]

LPC Solutions-Covariance Method

- now need to solve for \(\alpha\) using a 2-step procedure
  \[ VDV\alpha - \psi \]
  writing this as
  \[ YY' = \psi \] with
  \[ DV\alpha - Y \text{ or} \]
  \[ Y'\alpha = D^{-1}Y \]
  \(Y'\alpha\) from \(V\) (which is now known) solve for column vector \(Y\) using a simple recursion of the form
  \[
  Y_i = \psi_i - \sum_{j=1}^{i-1} Y_j Y_{ji}, \quad p \geq i \geq 2
  \]
  with initial condition
  \[ Y_1 = \psi_1 \]

Cholesky Decomposition Example

- continuing the example we solve for \(Y\)
  \[
  \begin{bmatrix}
  1 & 0 & 0 & 0 & Y_1 \\
  V_{21} & 1 & 0 & 0 & Y_2 \\
  V_{31} & V_{32} & 1 & 0 & Y_3 \\
  V_{41} & V_{42} & V_{43} & 1 & Y_4 \\
  \end{bmatrix}
  =
  \begin{bmatrix}
  \psi_1 \\
  \psi_2 \\
  \psi_3 \\
  \psi_4 \\
  \end{bmatrix}
  \]
  first solving for \(Y_1 - Y_4\) we get
  \[
  Y_1 = \psi_1 \\
  Y_2 = \psi_2 - V_{21}Y_1 \\
  Y_3 = \psi_3 - V_{31}Y_1 - V_{32}Y_2 \\
  Y_4 = \psi_4 - V_{41}Y_1 - V_{42}Y_2 - V_{43}Y_3 \\
  \]
Cholesky Decomposition Example

- next solve for $\alpha$ from equation

\[
\begin{bmatrix}
1 & V_{21} & V_{31} & V_{41} \\
0 & 1 & V_{22} & V_{32} \\
0 & 0 & 1 & V_{33} \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\alpha_4
\end{bmatrix}
= \begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3 \\
Y_4
\end{bmatrix}
\]

- giving the results

\[
\begin{align*}
\alpha_4 &= Y_4 / d_4 \\
\alpha_3 &= Y_3 / d_3 - V_3 / d_3 \alpha_4 \\
\alpha_2 &= Y_2 / d_2 - V_2 / d_2 \alpha_3 - V_2 / d_2 \alpha_4 \\
\alpha_1 &= Y_1 / d_1 - V_1 / d_1 \alpha_2 - V_1 / d_1 \alpha_3 - V_1 / d_1 \alpha_4
\end{align*}
\]

- completing the solution

Covariance Method Minimum Error

- the minimum mean squared error can be written in the form

\[
E_p = \phi_0(0,0) - \sum_{k=0}^{p-1} \phi(k,0)k^n
\]

- since $\alpha^T Y = Y^T V \alpha$ can write this as

\[
E_p = \phi_0(0,0) - Y^T D \alpha
\]

- this computation for $E_p$ can be used for all values of LP order from 1 to $p$ so can understand how LP order reduces mean-squared error

Solutions of LPC Equations

Autocorrelation Method via Levinson-Durbin Algorithm

Levinson-Durbin Algorithm 1

- Autocorrelation equations (at each frame $i$):

\[
\sum_{k=0}^{p} \alpha_k R[i-k] = R[i], \quad 1 \leq i \leq p
\]

\[
Ra = R
\]

- $R$ is a positive definite symmetric Toeplitz matrix

- The set of optimum predictor coefficients satisfy:

\[
R[i] - \sum_{k=0}^{p} \alpha_k R[i-k] = 0, \quad 1 \leq i \leq p
\]

- with minimum mean-squared prediction error of:

\[
E_p = \sum_{k=0}^{p} \alpha_k E_p
\]

Levinson-Durbin Algorithm 2

- By combining the last two equations we get a larger matrix equation of the form:

\[
\begin{bmatrix}
\vdots & \vdots & \vdots & \ddots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_p
\end{bmatrix}
= \begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

- expanded $(p+1) \times (p+1)$ matrix is still Toeplitz and can be solved iteratively by incorporating new correlation value at each iteration and solving for next higher order predictor in terms of new correlation value and previous predictor.
Levinson-Durbin Algorithm 3

Show how \( n \) order solution can be derived from \((n-1)^{st}\) order solution, i.e., given \( \alpha^{(n-1)} \), the solution to \( R^{(n)} \alpha^{(n)} = E^{(n)} \) we derive solution to \( R^{(n)} \alpha^{(n)} = E^{(n)} \).

The \((n-1)^{st}\) solution can be expressed as:

\[
\begin{bmatrix}
\end{bmatrix}
\begin{bmatrix}
1 \\
-R_1^{(n-1)} \\
-R_2^{(n-1)} \\
\vdots \\
-R_{n-1}^{(n-1)}
\end{bmatrix}
= E^{(n-1)}
\]

Levinson-Durbin Algorithm 4

Appending a 0 to vector \( \alpha^{(n)} \) and multiplying the matrix \( R^{(n)} \) gives new set of equations of the form:

\[
\begin{bmatrix}
\end{bmatrix}
\begin{bmatrix}
1 \\
-R_1^{(n-1)} \\
-R_2^{(n-1)} \\
\vdots \\
-R_{n-1}^{(n-1)}
\end{bmatrix}
= E^{(n-1)}
\]

where \( \gamma^{(n)} = \sum_{j=0}^{n} \alpha^{(n)} R[j] \) and \( R[i] \) are introduced.

Levinson-Durbin Algorithm 5

Key step is that since Toeplitz matrix has special symmetry we can reverse the order of the equations (first equation last, last equation first), giving:

\[
\begin{bmatrix}
\end{bmatrix}
\begin{bmatrix}
0 \\
1 \\
-R_1^{(n-1)} \\
-R_2^{(n-1)} \\
\vdots \\
-R_{n-1}^{(n-1)}
\end{bmatrix}
= E^{(n-1)}
\]

Levinson-Durbin Algorithm 6

To get the equation into the desired form (a single component in the vector \( E^{(n)} \)) we combine the two sets of matrices (with a multiplicative factor \( k_i \)) giving:

\[
\begin{bmatrix}
1 & 0 & 0 & \ldots & 0
\end{bmatrix}
\begin{bmatrix}
E^{(n-1)} \\
\gamma^{(n)}
\end{bmatrix}
= \begin{bmatrix}
E^{(n)}
\end{bmatrix}
\]

\[
k_i = \frac{\gamma^{(n)}}{R[0]} = \frac{\gamma^{(n)}}{E^{(n)}} - \sum_{j=0}^{n} \alpha^{(n)} R[j]
\]

Levinson-Durbin Algorithm 7

The first element of the right hand side vector is now:

\[
E^{(n)} = E^{(n-1)} - k_i \alpha^{(n)} = E^{(n-1)} (1 - k_i)
\]

The \( k_i \) parameters are called PARCOR coefficients.

With this choice of \( \gamma^{(n)} \), the vector of \( n \) order predictor coefficients is:

\[
\begin{bmatrix}
1 \\
\alpha_1^{(n)} \\
\alpha_2^{(n)} \\
\vdots \\
\alpha_{n-1}^{(n)} \\
0
\end{bmatrix}
= \begin{bmatrix}
1 \\
-R_1^{(n-1)} \\
-R_2^{(n-1)} \\
\vdots \\
-R_{n-1}^{(n-1)} \\
0
\end{bmatrix}
\]

yielding the updating procedure:

\[
\begin{align*}
\alpha_0^{(n)} &= \alpha_0^{(n-1)} - k_i \alpha_{n-1}^{(n-1)}\quad j=1,2,\ldots
\end{align*}
\]

\[
\alpha_i^{(n)} = k_i
\]

Levinson-Durbin Algorithm 7

The final solution for order \( p \) is:

\[
a_j = \alpha_j^{(p)}, \quad 1 \leq j \leq p
\]

with prediction error

\[
E^{(p)} = E(0) \prod_{k=0}^{p} (1-k_k^2) = R(0) \prod_{k=0}^{p} (1-k_k^2)
\]

If we use normalized autocorrelation coefficients:

\[
r[k] = R[k] / R[0]
\]

we get normalized errors of the form:

\[
\nu^{(p)} = \frac{E^{(p)}}{R[0]} = 1 - \sum_{j=0}^{p} \alpha^{(p)} r[j] = \prod_{k=0}^{p} (1-k_k^2)
\]

where \( 0 < \nu^{(p)} \leq 1 \) or \(-1 < k_k < 1\)
Levinson-Durbin Algorithm

Let $\varepsilon^{(i)} = R[i]$ for $i = 1, 2, \ldots, p$.

$$k_i = \left( R[i] - \frac{1}{p} \sum_{j=1}^{p} a_j^{(i-1)} R[j-1] \right) \varepsilon^{(i-1)}$$

(9.98)

$$a_j^{(i)} = k_i$$

if $i > 1$ and $j = 1, 2, \ldots, i - 1$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}$$

(9.99)

End

$$\varepsilon^{(i)} = (1 - k_i^2) \varepsilon^{(i-1)}$$

(9.10)

End

$$\alpha_j = \alpha_j^{(p)} \quad j = 1, 2, \ldots, p$$

(9.11)

Autocorrelation Example

Consider a simple $p = 2$ solution of the form

$$\begin{bmatrix} R(0) & R(1) \\ R(1) & R(2) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \end{bmatrix}$$

With solution

$$E^{(0)} = R(0)$$

$$k_2 = R(1)/R(0)$$

$$\alpha_2^{(1)} = R(1)/R(0)$$

$$E^{(1)} = \frac{R^2(0) - R^2(1)}{R(0)}$$

Prediction Error as a Function of $p$

$$V_i = \frac{E_i}{R_i[0]} = 1 - \sum_{k=1}^{i} \frac{R_k[k]}{R_k[0]}$$

Model order is usually determined by the following rule of thumb:

- $F_s/1000$ poles for vocal tract
- 2-4 poles for radiation
- 2 poles for glottal pulse

Autocorrelation Method Properties

- Mean-squared prediction error always non-zero
  - Decreases monotonically with increasing model order
- Autocorrelation matching property
  - Model and data match up to order $p$
- Spectrum matching property
  - Favors peaks of short-time FT
- Minimum-phase property
  - Zeros of $A(z)$ are inside the unit circle
- Levinson-Durbin recursion
  - Efficient algorithm for finding prediction coefficients
  - PARCOR coefficients and MSE are by-products