

Signal Processing Techniques and Its Applications
Dr. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 47
Covariance Method for Linear Prediction

Ok. So, in the last class, we talk about the calculation of α value using the autocorrelation function. So, autocorrelation I have explained all those things in autocorrelation.

(Refer Slide Time: 00:30)

There are $L-|i-k|$ non-zero terms in the computation of $\phi_n[i, k]$ for each value of i, k

Then $\phi_n[i, k] = R_n[i - k]$ short-time autocorrelation

$$R_n[k] = \sum_{m=0}^{L-1-k} S_n[m] S_n[m+k]$$


From equation (2)

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

$$\sum_{k=1}^p \alpha_k R_n[i - k] = R_n[i] \quad 1 \leq i \leq p$$

Minimum mean-squared prediction error can be written as

$$E_n = \phi_n[0, 0] - \sum_{k=1}^p \alpha_k \phi_n[0, k]$$

$$= R_n[0] - \sum_{k=1}^p \alpha_k R_n[k]$$


So, we are here to say that we have derived this equation. So, what is the equation? We said that the equation is this one α . This one is equal to $R_n i$. Now, if I want to represent this set of equations in matrix form, I can say that k varies from 1 to p , so let us say i equals 0. So, i equals 0 minus 0 minus k equals 1, so minus 1.

(Refer Slide Time: 00:56)

$$\begin{bmatrix} R_n[0] & R_n[1] & R_n[2] & \dots & R_n[p-1] \\ R_n[1] & R_n[0] & R_n[1] & \dots & R_n[p-2] \\ R_n[2] & R_n[1] & R_n[0] & \dots & R_n[p-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n[p-1] & R_n[p-2] & R_n[p-3] & \dots & R_n[0] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n[1] \\ R_n[2] \\ R_n[3] \\ \vdots \\ R_n[p] \end{bmatrix}$$

$R_n \alpha = r$
 $\alpha = R_n^{-1} r$
 $R_n = \frac{SNL}{EN}$

This is a $p \times p$ Toeplitz Matrix \Rightarrow symmetric with all diagonal elements equal
matrix equation solved using Levinson or Durbin method

So, I can say that $R_0, R_1, R_2, \dots, R_{p-1}$, which is nothing but a multiplied by the α_1 equal to 1, means α_1 . So, α_1 , then it is nothing but a 1. So, I equal to I said I equal to 1. So, 1 minus 1 is equal to 0. So, it is R_0 , then say that 1 minus 2. So, mod of minus 1; that means mod means 1.

So, this is R_1 , then R_2, \dots, R_{p-1} . And that is equal to multiplying by α_1 , and that is R_n . I equal to 1, then I equal to 2, I equal to 3, I equal to 4. I will get that, and that is the matrix form of this equation. So, I can say this matrix so, this is matrix let us say capital R ; capital R into α equal to small r . What is capital R ? It is nothing but an auto-correlation matrix.

R_{n0} is nothing but a signal multiplied by the same signal, which is nothing but the energy; R_{n1} just shifted by one sample. So, if I know all R , if I know this R_{n1}, R_{n2} , this one, then I can easily calculate the α value. So, this is the procedure and that procedure is called the Levinson Durbin method. So, this matrix is solved using Levinson Durbin methods.

(Refer Slide Time: 02:47)

The set of optimum predictor coefficients satisfy

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

with minimum mean-squared prediction error of

$$R_n[0] - \sum_{k=1}^p \alpha_k R_n[k] = E^{(p)}$$

Now, another one is that the minimum mean square error E_n is equal to $\phi_n^T \mathbf{0}$. So, I can say that minimum mean square error can be expressed using the same procedure now if I want to represent this one in a matrix form.

(Refer Slide Time: 02:56)

$$\begin{bmatrix} R_n[0] & R_n[1] & R_n[2] & \dots & R_n[p] \\ R_n[1] & R_n[0] & R_n[1] & \dots & R_n[p-1] \\ R_n[2] & R_n[1] & R_n[0] & \dots & R_n[p-2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n[p-1] & R_n[p-2] & R_n[p-3] & \dots & R_n[0] \end{bmatrix} \begin{bmatrix} 1 \\ -\alpha_1^{(p)} \\ -\alpha_2^{(p)} \\ \vdots \\ -\alpha_p^{(p)} \end{bmatrix} = \begin{bmatrix} E^{(p)} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Expanded matrix is still Toeplitz and It 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

i^{th} order solution can be derived from $(i-1)^{\text{st}}$ order solution

given $\alpha^{(i-1)}$, the solution to

we derive solution to $R_n^{(i-1)} \alpha^{(i-1)} = E_n^{(i-1)}$

So, I can say this is nothing but R_0 , R_1 , and R_2 multiplied by 1. I get R_0 , which is equal to E_p . The Rest are minus α_1 , minus α_2 , minus α_3 I multiplied and which are equal to 0.

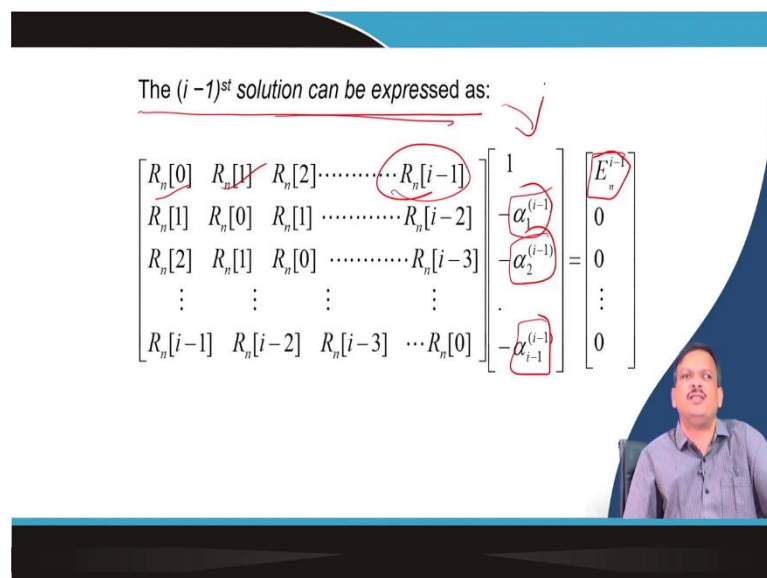
That means when my predictor is 100 per cent correct, then the error is equal to 0 OKs. Now, I can solve this matrix iteratively. What are the iterative methods? That means the

ith order solution can be derived from $i-1$ order solution. So, physically what we are doing, let us say we are calculating the error for a first set of α ; $\alpha_1, \alpha_2, \alpha_p$. Then I know that when I am predicting nothing to a signal, the prediction error is maximum.

Now, when I go, the order increases. So, when I say the 1st order, I am predicting, that means I am predicting all 0th samples and the 1st sample, which will be an error, will be maximum, and when the order is gone, then the error will be minimum. So, I can say that I can say that I 2nd order prediction can be derived from 1st-order prediction. So, $R_n^{i-1} \alpha_i$ minus 1 is equal to E_n^{i-1} ok.

(Refer Slide Time: 04:35)

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

$$\begin{bmatrix} R_n[0] & R_n[1] & R_n[2] & \dots & R_n[i-1] \\ R_n[1] & R_n[0] & R_n[1] & \dots & R_n[i-2] \\ R_n[2] & R_n[1] & R_n[0] & \dots & R_n[i-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n[i-1] & R_n[i-2] & R_n[i-3] & \dots & R_n[0] \end{bmatrix} \begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \end{bmatrix} = \begin{bmatrix} E_n^{(i-1)} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



So, now, if I say that, then I can say, what is minus 1? What is a solution? R_0, R_1, R^{i-1} . So, it is nothing but an $E_n^{i-1} \alpha_1^{i-1} \alpha_2^i$. So, 1st set of predictions, then 2nd iteration, then 3rd iteration. So, once the iteration is going on, that error will be minimized.

(Refer Slide Time: 05:07)

Appending a 0 to vector α and multiplying by the matrix $R_n^{(i)}$ gives

$$\begin{bmatrix} R_n[0] & R_n[1] & R_n[2] & \dots & R_n[i] \\ R_n[1] & R_n[0] & R_n[1] & \dots & R_n[i-1] \\ R_n[2] & R_n[1] & R_n[0] & \dots & R_n[i-2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n[i-1] & R_n[i-2] & R_n[i-3] & \dots & R_n[1] \\ R_n[i] & R_n[i-1] & R_n[i-2] & \dots & R_n[0] \end{bmatrix} \begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \\ 0 \end{bmatrix} = \begin{bmatrix} E_n^{i-1} \\ 0 \\ 0 \\ \vdots \\ 0 \\ \gamma^{(i-1)} \end{bmatrix}$$


where $\gamma^{(i-1)} = R_n[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R_n[i-j]$



Now, I can say that this is the iterative method that I used to solve this matrix. So, if I solve this matrix, I just show you that because this is mathematics, you can see that matrix solution by going through the books. So, mathematics we can give. So, now, I append 0 to a vector α and multiply by R_n^{i-1} .

(Refer Slide Time: 05:24)

Toeplitz matrix has special symmetry we can reverse the order of the equations

$$\begin{bmatrix} R_n[0] & R_n[1] & R_n[2] & \dots & R_n[i] \\ R_n[1] & R_n[0] & R_n[1] & \dots & R_n[i-1] \\ R_n[2] & R_n[1] & R_n[0] & \dots & R_n[i-2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n[i-1] & R_n[i-2] & R_n[i-3] & \dots & R_n[1] \\ R_n[i] & R_n[i-1] & R_n[i-2] & \dots & R_n[0] \end{bmatrix} \begin{bmatrix} 0 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma^{(i-1)} \\ 0 \\ 0 \\ \vdots \\ 0 \\ E_n^{i-1} \end{bmatrix}$$


So, I am append 0 and multiplying by R_n^{i-1} . I get this one, sorry, so, 1 0. Now, appending 0 in here. So, if this is multiplied by column 0 0, nothing will happen. So, I can multiply this one and append a 0 vector here, which is nothing but a σ^{i-1} ; let us use this side.

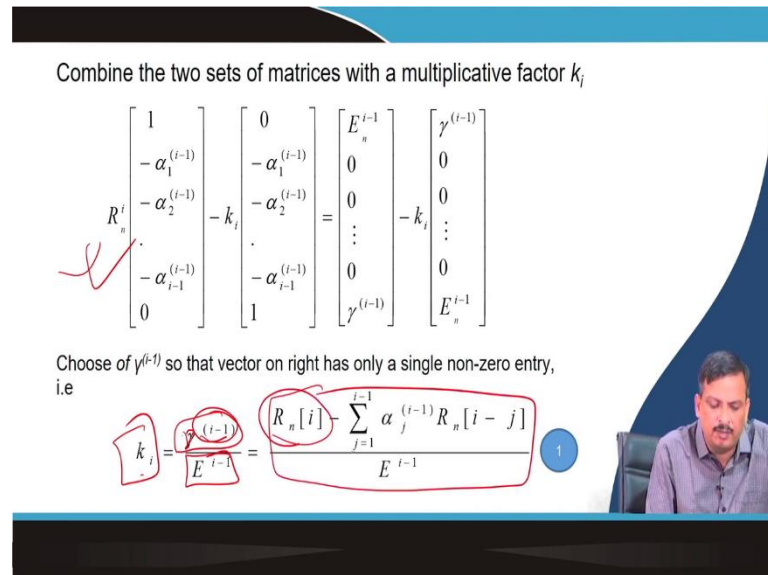
So, this can be solved from this matrix: $\sigma I - \psi^{i-1}$ is equal to $R_n I - j$ equal to 1 to $i-1$ this one, then I can reverse because it is a symmetry matrix.

(Refer Slide Time: 06:17)

Combine the two sets of matrices with a multiplicative factor k_i

$$R_n^i \begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \\ 0 \end{bmatrix} - k_i \begin{bmatrix} 0 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \\ 1 \end{bmatrix} = \begin{bmatrix} E_n^{i-1} \\ 0 \\ 0 \\ \vdots \\ 0 \\ \gamma^{(i-1)} \end{bmatrix} - k_i \begin{bmatrix} \gamma^{(i-1)} \\ 0 \\ 0 \\ \vdots \\ 0 \\ E_n^{i-1} \end{bmatrix}$$

Choose of $\gamma^{(i-1)}$ so that vector on right has only a single non-zero entry, i.e

$$k_i = \frac{\gamma^{(i-1)}}{E_n^{i-1}} = \frac{R_n[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R_n[i-j]}{E_n^{i-1}}$$


So, whether it is reversed, the order does not change it. So, I can change the reverse order, and then I can write in this combination. From this combination, I can say that k_i is equal to $\sigma \psi^{i-1}$. This is, sorry, γ^{i-1} divided by E^{i-1} . So, γ^{i-1} I have already done that divided by E^{i-1} , I have done that. So, I can say k_i is equal to $R_n[i]$ equal to 1 to $i-1$. So, this is one equation I can get.

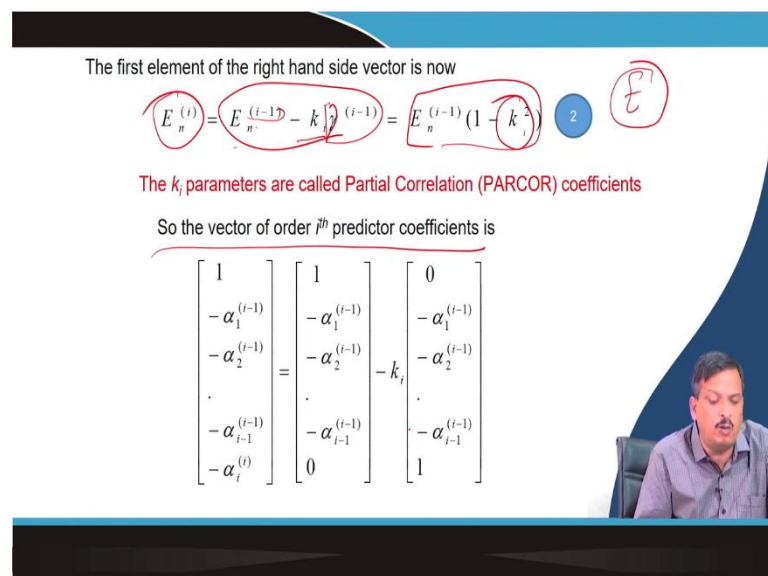
(Refer Slide Time: 06:50)

The first element of the right hand side vector is now

$$E_n^{(i)} = E_n^{(i-1)} - k_i \gamma^{(i-1)} = E_n^{(i-1)} (1 - k_i \gamma^{(i-1)})$$

The k_i parameters are called Partial Correlation (PARCOR) coefficients

So the vector of order i^{th} predictor coefficients is

$$\begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \\ -\alpha_i^{(i)} \end{bmatrix} = \begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \\ 0 \end{bmatrix} - k_i \begin{bmatrix} 0 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ \vdots \\ -\alpha_{i-1}^{(i-1)} \\ 1 \end{bmatrix}$$


Now, once I get the k_i , I can calculate that E_n , E_{n+1} is nothing but an E_{n-1} minus k_i^2 this γ^{i-1} . So, it is nothing but an E_{n-1} minus k_i^2 . So, I can say the energy once the prediction order is increased iteration increased energy will be minimized by multiplying by k_i^2 and that k_i is called the partial correlation coefficient or PARCOR. So, k_i is called the partial correlation coefficient, then I solve for i th predictor, and I can solve for α_j .

(Refer Slide Time: 07:34)

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$$

$$\alpha_i^{(i)} = k_i$$

A. The final solution for order p is:

$$\alpha_j = \alpha_j^p \quad 1 \leq j \leq p$$

B. with prediction error

$$E_n^{(p)} = E_n[0] \prod_{m=1}^p (1 - k_m^2) = R_n[0] \prod_{m=1}^p (1 - k_m^2)$$

C. If we use normalized autocorrelation coefficients:

$$r_n[k] = \frac{R_n[k]}{R_n[0]}$$

D. normalized prediction error

$$v^{(i)} = \prod_{m=1}^i (1 - k_m^2) \quad 0 \leq v^{(i)} \leq 1 \quad -1 \leq k_i \leq 1$$

So, ultimately, I can say there are four equations I have to solve for calculating the α value. 1st equation is this one. So, the 1st equation is this one, the k_i is equal to this one, the 2nd equation is this one, the 3rd equation is this one and the 4th equation is this one.

(Refer Slide Time: 08:18)

Handwritten mathematical derivations for the Levinson-Durbin algorithm:

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

for $i = 1, 2, \dots, p$

$$k_i = \frac{R[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R[i-j]}{E^{(i-1)}}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$$

$$\alpha_i^{(i)} = k_i$$

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

end

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

Additional notes on the right:

$$R_k = \frac{\gamma^{(i+1)} - \sum_{j=1}^k R_{i-j+1} R_{i-j}}{E^{(i-1)}}$$

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

Small diagram showing the relationship between R , E , and α .

So, let us say I want to implement it using a program. So, what can I do? I can calculate that. Let us say I copy these whole four equations on one page. Now, let us say I want to implement these four equations in a program. So, what is required if you see what is required from the 1st equation? From the 1st equation, what is the requirement? If I want to calculate k_1 , let us say i is equal to 1. So, I require R_1 , I require E_0 , I require R .

So, i equal to 1. So, i equal to 1 means j is equal to 1 to 1 minus 1. So, j is equal to so, 1 to 0. So, that term will not be there. So, I require k_1 to calculate k_1 , and I require R_1 and E_0 . So, what is R_1 ? What is R_1 ? R_1 is nothing but an autocorrelation coefficient shifted by 1. So, I can suppose I have given a signal. Let us say I give you a signal that I have a speech signal.

Let us say the speech signal is recorded with an 8-kilo hertz sampling frequency, and then I take a window size of 20 milliseconds. So, I have 160 samples in my speech signal. So, from 160 samples, I have to calculate this R_0, R_1, R_2 autocorrelation coefficient. So, what is the equation? I know R is R_k is equal to k equal, so let us say n equal to 0 to n minus capital N minus 1 or capital L minus 1. This is L , $x[n]$ multiplied by again $x[n+k]$. Let us say.

Outside the window L , the signal is 0. So, when I say R_0 , that means k is equal to 0; that means I am multiplying $x[n]$ with $x[n]$. So, each sample will multiply its own sample. So,

it is nothing but x square of n , so that means nothing but the energy of the signal. So, it is nothing but an E_0 . So, if R_0 is equal to E_0 , then what is R_1 ?

R_1 is nothing but a shift of the signal by one sample $x[n]$ multiplied by $x[n]$ plus 1, R_1 I can calculate. So, let us say I calculate R_0 and R_1 . Then I know what k_i ? k_i is i is nothing but an R_0 ; R_0 I know my divided by so, R_1 I know I divided by E_0 . So, E_0 , I know R_0 is equal to E_0 . So, k_1 , I know. So, similarly, if I want to calculate k_2 , then you can see I required R_2 , I required R_1 , and I required E_0 .

So, I require R_2 , R_1 , and E_0 , and then I will use the same equation I am implementing. So, I will vary from 1 to p , and k I will be k_1 and k_2 to k_p . Now, once I calculate k_1 , this for loop will be executed for i equal to 1 to p ok. Then I know α_j is equal to k_i ok. Then, if I see this loop, then I can calculate E_{n_i} because when I say k_2 , I require E_1 . What is E_1 ?

E_1 is nothing but an E_0 minus k_1 square $1 - k_1$ square into E_0 . E_0 I know k_1 I know. So, I can calculate E_1 ; E_1 is required for k_2 calculation α_i . So, α_1 is equal to k_1 . So, i equal to 1 to 2. So, α_1 2, α_3 1. So, iteration is 1 first iteration. When I said, then I equal to 1 to p . So, in the first iteration, I get α_1 1 α_1 2. So, α_2 1 α_3 1.

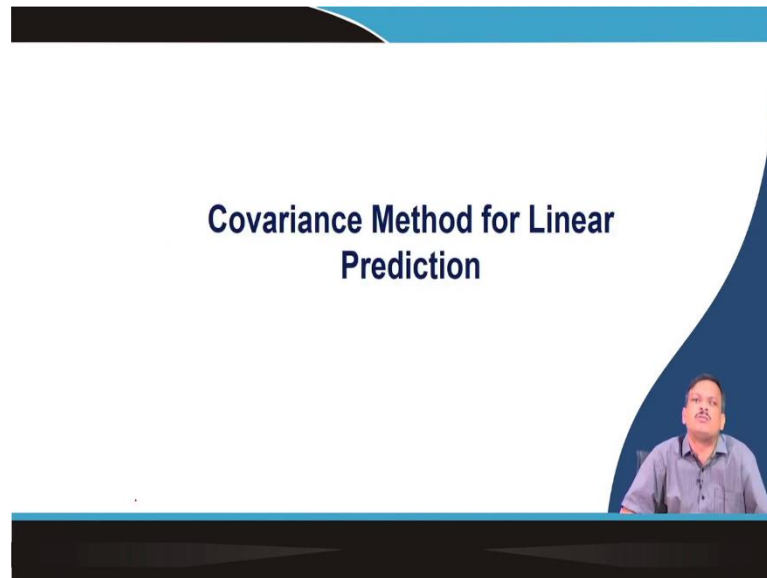
So, the upper one is the iteration, and the lower one is the α different α value. Now, once I get that, then I know if I am equal to 1, then nothing is there; if I am equal to 1, but if the because it is α_1 0. So, more than 1. So, j is equal to 1 to $i-1$ when I say, then I can say. So, α_1 I cannot iterate. So, from the second stage, I can iterate.

So, I can say the j is equal to 1 α_2 let us say. So, j equal to 1 α_2 is nothing but a j equal to 1 α_1 minus k_1 into α_1 1 understand and modify the energy. So, when I equal to 1, this portion will not enter. So, it modifies the energy and goes back to here, and for i equal to 2, the k_2 will be calculated, then the α_1 1 will be 1 2 will be calculated.

That way, this whole equation will be implemented. So, I ultimately get α_1 p . So, what I am saying is that I am calculating α value based on the iteration value. So, if you use this for loop implement in a C program and check for any recording of sound, let us say record your name and implement it take a frame and then implement it to calculate that α value. Outside the window, your signal is 0.

First, you have to calculate all autocorrelation coefficients, and then you have to do it. So, that is the one method for the calculation of the linear prediction coefficient using the autocorrelation method. What is the assumption? Outside the window, the signal is 0, and the error is maximum at the beginning of the window; at the end, the error is maximum ok.

(Refer Slide Time: 15:14)



(Refer Slide Time: 15:19)

Can find values of α_k that minimize by setting

$$\frac{\partial E_n}{\partial \alpha_i} = 0, \quad i = 1, 2, \dots, p$$

$$\frac{\partial E_n}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_i} \sum_{m=-\infty}^{\infty} (s_n[m] - \sum_{k=1}^p \alpha_k s_n[m-k])^2$$

$$= 2 \sum_{m=-\infty}^{\infty} (s_n[m] - \sum_{k=1}^p \alpha_k s_n[m-k]) \left(-\frac{\partial}{\partial \alpha_i} \sum_{k=1}^p \alpha_k s_n[m-k] \right)$$

Where

$$-s_n[m-i] = -\frac{\partial}{\partial \alpha_i} \sum_{k=1}^p \alpha_k s_n[m-k]$$

$\alpha_k s_n[m-k]$ is constant with respect to $\frac{\partial}{\partial \alpha_i}$ for

The slide continues the derivation from the previous slide. It shows the partial derivative of the error energy E_n with respect to the coefficient α_i set to zero. It then expands the derivative using the chain rule, showing that the derivative of the squared error term is twice the error term multiplied by the negative derivative of the prediction sum. The final line indicates that terms where $k \neq i$ are constant with respect to α_i .

(Refer Slide Time: 15:22)

$$0 = 2 \sum_{n=-\infty}^{\infty} (s_n[m] - \sum_{k=1}^p \alpha_k s_n[m-k]) (-s_n[m-i])$$

$$\sum_{n=-\infty}^{\infty} s_n[m-i] s_n[m] = \sum_{k=1}^p \alpha_k \sum_{n=-\infty}^{\infty} s_n[m-i] s_n[m-k] \quad 1 \leq i \leq p \quad (1)$$


let

$$\phi_n[i, k] = \sum_m s_n[m-i] s_n[m-k] \quad 1 \leq i \leq p$$

then

$$\phi_n[i, 0] = \sum_{k=1}^p \alpha_k \phi_n[i, k] \quad i = 1, 2, \dots, p \quad (2)$$

leading to a set of p equations in p unknowns that can be solved in an efficient manner for the $\{\alpha_k\}$



(Refer Slide Time: 15:27)

$$\phi_n[i, k] = \sum_m s_n[m-i] s_n[m-k] \quad 1 \leq i \leq p$$

$$\phi_n[i, k] = \sum_{m=0}^{L-i} s_n[m-i] s_n[m-k] \quad 1 \leq i \leq p \quad 0 \leq k \leq p$$


Changing the summation index gives

$$\phi_n[i, k] = \sum_{m=-i}^{L-i-1} s_n[m] s_n[m+i-k] \quad 1 \leq i \leq p \quad 0 \leq k \leq p$$

$$\phi_n[i, k] = \sum_{m=-k}^{L-k-1} s_n[m] s_n[m+k-i] \quad 1 \leq i \leq p \quad 0 \leq k \leq p$$

key difference from Autocorrelation Method is that limits of summation include terms before $m=0$. i.e window extends p samples backwards

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



(Refer Slide Time: 15:28)

$$\phi_n[i,0] = \sum_{k=1}^p \alpha_k \phi_n[i,k] \quad i = 1, 2, \dots, p$$

$$E_n = \phi_n[0,0] - \sum_{k=1}^p \alpha_k \phi_n[0,k]$$

$$\begin{bmatrix} \phi_n[1,1] & \phi_n[1,2] & \phi_n[1,3] & \dots & \phi_n[1,p] \\ \phi_n[2,1] & \phi_n[2,2] & \phi_n[2,3] & \dots & \phi_n[2,p] \\ \phi_n[3,1] & \phi_n[3,2] & \phi_n[3,3] & \dots & \phi_n[3,p] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_n[p,1] & \phi_n[p,2] & \phi_n[p,3] & \dots & \phi_n[p,p] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} \phi_n[1,0] \\ \phi_n[2,0] \\ \phi_n[3,0] \\ \vdots \\ \phi_n[p,0] \end{bmatrix}$$

$$\phi \alpha = \psi$$

$$\alpha = \phi^{-1} \psi$$

Now, there is another method called the covariance method. instead of going to autocorrelation, I can directly calculate. So, the ϕ matrix is the same. So, this is the derivation of the ϕ matrix, which is the same here also. So, once I get this ϕ matrix. What is ϕ matrix? It is nothing but a covariance.

So, instead of replacing the covariance with an autocorrelation coefficient, I can directly calculate $\phi[1, 1]$, I can directly calculate $\phi[1, 2]$. I am saying the signal is not 0 outside the window. So, I am not taking the signal as 0 outside the window.

I am taking. I am saying I have a signal. I take a window. So, outside the window, the signal is not 0. So, when I calculate the error prediction error for this sample, I do not predict from 0; I predict from the original sample value. So, that error is not maxima at the boundary edge. So, I directly calculate all the covariance. So, then, that means I know the ϕ matrix. I so, once I know the ϕ matrix, I can calculate the α matrix by simple matrix decomposition method.

(Refer Slide Time: 16:37)

the solution of the matrix equation is called the Cholesky decomposition, or square root method

ϕ Matrix is symmetric but not diagonal elements are same

$$\phi = ADA^T$$

Where A = lower triangular matrix with 1's on the main diagonal; D=diagonal matrix

determine elements of A and D by solving for (i, j) elements of the matrix equation



What is the matrix decomposition method? Let us say I have a matrix ϕ is a matrix symmetric matrix because if you see $\phi[1, 1]$, $\phi[1, 2]$. So, all diagonal elements $\phi[1, 1]$, $\phi[2, 2]$, $\phi[p, p]$ ok. So, all diagonal elements are the same; that means $\phi[1, 1]$ is not the same, but there are diagonal elements are there.

And those are the, but if you see this portion of the matrix and this portion of the matrix is same. So this is called the upper triangular matrix; this is called the lower triangular matrix.

(Refer Slide Time: 17:24)

Let $p=4$ and matrix element $\phi_n(i, j) = \phi_{ij}$

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ A_{21} & 1 & 0 & 0 \\ A_{31} & A_{32} & 1 & 0 \\ A_{41} & A_{42} & A_{43} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_3 & 0 \\ 0 & 0 & 0 & d_4 \end{bmatrix} \begin{bmatrix} 1 & A_{21} & A_{31} & A_{41} \\ 0 & 1 & A_{32} & A_{42} \\ 0 & 0 & 1 & A_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Handwritten notes: ϕ_{ij} , d_1, d_4



So, I can say that let us say, for example, let us say I have a 4th order 1 2 prediction 4th L α_1 to α_4 α_1 2 3 4 I want to predict. So, what do I know? I know $\phi[1, 1]$, $\phi[2, 1]$, $\phi[3, 1]$, $\phi[4, 1]$, $\phi[2, 1]$, $\phi[2, 2]$, $\phi[3, 2]$, $\phi[4, 2]$ understand.

So, now, I know all those things. So, I can say that all one is a diagonal element, one is an upper triangular matrix, and one is a lower triangular matrix. So, let us say I decomposed this matrix into a lower triangular matrix, diagonal matrix and upper triangular matrix. If I multiply these three matrices, I will get this one. Now, if you see that if I do that, if I solve this matrix. So, I can say that d_1 . So, first, you multiply this one by this column.

So, d_1 then this the I can say that this column with this row. So, I can say d_1 is equal to $\phi[1, 1]$ only else are 0.

(Refer Slide Time: 18:47)

Solve matrix

$$d_j = \phi_{jj} - \sum_{k=1}^{j-1} A_{jk}^2 d_k$$

$$A_{ji} = \frac{\phi_{ji} - \sum_{k=1}^{j-1} A_{jk} d_k A_{ki}}{d_j}$$

$$d_p = \phi_{pp} - \sum_{k=1}^{p-1} A_{pk}^2 d_k$$

Handwritten notes and boxed formulas on the slide:

- $d_1 = \phi_{11}$
- $A_{21} d_1 = \phi_{21} \Rightarrow A_{21} = \frac{\phi_{21}}{d_1}$
- $d_2 = \phi_{22} - (A_{21})^2 d_1$
- $A_{31} d_1 = \phi_{31} \Rightarrow A_{31} = \frac{\phi_{31}}{d_1}$
- $A_{41} d_1 = \phi_{41} \Rightarrow A_{41} = \frac{\phi_{41}}{d_1}$
- $A_{32} d_2 = \phi_{32} - A_{31} d_1 A_{21} \Rightarrow A_{32} = \frac{\phi_{32} - A_{31} d_1 A_{21}}{d_2}$
- $A_{42} d_2 = \phi_{42} - A_{41} d_1 A_{21} \Rightarrow A_{42} = \frac{\phi_{42} - A_{41} d_1 A_{21}}{d_2}$

Iterate procedure to solve for d_3, A_{43}, d_4

```

for(j=2; j<=p; j++)
{
    d_j = phi_jj - sum_{k=1}^{j-1} A_jk^2 d_k
    for(i=j+1; i<=p; i++)
    {
        A_ji = (phi_ji - sum_{k=1}^{j-1} A_jk d_k A_ki) / d_j
    }
}

```

So, $\phi[1, 1]$ is equal to d_1 . Then what is d_2 ? d_2 is nothing but a $\phi[2, 2]$, d_2 is nothing but a $\phi[2, 2]$ minus A_{21} square into d_1 . And if you see $A_{21} d_1$ is equal to $\phi[2, 1]$ $A_{21} d_1$ come in here. A_{21} multiplied by d_1 is equal to $\phi[2, 1]$ ok. So, I can say that $\phi[2, 1]$.

So, I can say A_{21} is nothing but a $\phi[2, 1]$ divided by d_1 . So, I know d_1 I know $\phi[2, 1]$. So, I can calculate A_{21} , I can calculate A_{31} , I can calculate A_{41} . Since a d_1 is known and all ϕ is known. So, first step, what is the first step? The first one, d_1 , is equal to $\phi[1, 1]$. This is the generalization.

Rest cases I can say that A. So, if I generalize this one, A_{21} , so, 2 1 means, let us say, A i 1. So, this is 1, which is fixed; only say the 2, 3, and 4 are varying. So, i from A11 is equal to d 1. So, diagonal, I can say that ϕ_i is equal to 1 2. So, $A_{\phi[1, 1]}$ divided by d 1. So, I can say that A_{21} . So, I can say that ϕ_{i1} is equal to ϕ_{i1} divided by d 1 else. Only the case of A11 is equal to $\phi[1, 1]$ divided by d 1.

Else I can say that $A_{ij} - A_{i1} A_{1j} / A_{11}$ is nothing but a ϕ_{ij} minus this one. So, first you just do for this one and then you go for if you see that A_{32} , A_{42} I can calculate like this one. So, I just implemented that in C programming. So, once I know the d p A matrix and the d matrix I can, I can calculate it from there from a given ϕ matrix.

(Refer Slide Time: 21:34)

$ADA^T \alpha = \psi$
 Let $AY = \psi$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ A_{21} & 1 & 0 & 0 \\ A_{31} & A_{32} & 1 & 0 \\ A_{41} & A_{42} & A_{43} & 1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix}$$

$$Y_1 = \psi_1$$

$$Y_2 = \psi_2 - A_{21} Y_1$$

$$Y_3 = \psi_3 - A_{31} Y_1 - A_{32} Y_2$$

$$Y_4 = \psi_4 - A_{41} Y_1 - A_{42} Y_2 - A_{43} Y_3$$

$$Y_i = \psi_i - \sum_{j=1}^{i-1} A_{ij} Y_j$$

```

Y1=psi1
for(i=1; i<=p; i++)
{
    Y_i = psi_i - sum_{j=1}^{i-1} A_ij Y_j
}
    
```

So, A and d I know. Let us say this: A matrix multiplied by Y is equal to this one, ok? So, now let us say this one multiplied by this one is equal to this one. Then you generalized it if I multiplied it: Y_1 is equal to this one, Y_2 is equal to this one, Y_2 this one minus this one, Y_1 . So, what is Y_1 ? I know what it is Y, then I can calculate Y_2 because A_{21} I already know. So, A_{21} is known as Y_1 . So, I can calculate Y_2 , Y_3 , Y_4 . So, I write down the program.

(Refer Slide Time: 22:22)

$$\Rightarrow D A' \alpha = Y \Rightarrow A' \alpha = D^{-1} Y$$

$$\begin{bmatrix} 1 & A_{21} & A_{31} & A_{41} \\ 0 & 1 & A_{32} & A_{42} \\ 0 & 0 & 1 & A_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 1/d_1 & 0 & 0 & 0 \\ 0 & 1/d_2 & 0 & 0 \\ 0 & 0 & 1/d_3 & 0 \\ 0 & 0 & 0 & 1/d_4 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix}$$

$$\alpha_4 = \frac{Y_4}{d_4}$$

$$\alpha_3 = \frac{Y_3}{d_3} - A_{43} \alpha_4$$

$$\alpha_2 = \frac{Y_2}{d_2} - A_{43} \alpha_4 - A_{32} \alpha_3$$

$$\alpha_1 = \frac{Y_1}{d_1} - A_{43} \alpha_4 - A_{32} \alpha_3 - A_{21} \alpha_2$$

$$\alpha_i = \frac{y_i}{d_i} - \sum_{j=i+1}^p A_{ji} \alpha_j$$

Calculation proceeds backwards from $i=p-1$ to 1

Now, like once I know that, what can I do? I can say the $A D$, $D A$ transpose because if you see that if the A this one is A , what is A transpose? Nothing but an upper triangular matrix. So, A transpose α is equal to D inverse Y . So, what is D ? D is nothing but a diagonal matrix. So, what is the inverse of the diagonal matrix? This one multiplied by Y is equal to A transpose α . Now, I calculate for α value α_4 .

So, if you see that order, I get α_4 first, α_3 second, α_2 third and α_1 first. So, it is in reverse order. So I can write down the program in reverse order. Maybe the programming will be reverse-ordered. Now, instead of forwarding the order, it will be reversed. Not forward order. This program is wrong. This is in forward order. It will be I equal to $p-2$ I less than p plus and $p-1$ minus.

So, then I get α_4 ; α_4 is equal to Y_4 divided by d_4 . Y_4 I knew from the previous slide and d_4 I can calculate here d_4 ; d_4 I know. So, I can calculate α_4 then α_3 , α_2 , α_1 this is known, this is known, this is known. So, this is a mistake; here is a mistake: if you see this for loop, it will be $p-i$ equal to $p-2$ less than $p-i$ minus less than equal to $p-i$ minus minus.

So, this is the matrix decomposition form. I can calculate that α value using matrix decomposition form, but what do I have to calculate? I have to calculate covariance. Is that okay or not? But, if you see in the case of autocorrelation, what do I have to calculate? I have to calculate the R matrix and the autocorrelation coefficient matrix. In the case of

autocorrelation, I have to calculate the R matrix. In the case of covariance, I have to calculate the ϕ matrix.

So, I can say the autocorrelation and covariance methods are two-stage method; in the first stage, I have to calculate autocorrelation case of the first stage, I have to calculate covariance. In the case of autocorrelation, I have to learn the Levinson recursion to find out the α value. In the case of covariance, I have to do the matrix decomposition to compute that α value.

So, in both cases, the second stage is iterative, and the first step is a calculation of the R -value and ϕ value. So, in both cases, what is the computational complexity in the case of autocorrelation methods and covariance methods? For the autocorrelation method in the first stage, I have to compute the autocorrelation, which requires computational complexity. Similarly, in covariance methods, I have to compute that covariance. So, that requires a computation.

So, if the L is equal to 160 samples, I have to calculate R_0 R . So, if I want to calculate the 16th order LPC, then L will be there every time, and p will be 1. So, 16 into L number of multiplication I have to do. In covariance, we also have to window length by that time. So, in both cases, computational complexity is in two stages: the first stage is the calculation of autocorrelation, the second stage is the calculation of the covariance, and the second stage is the iteration.

So, now, people are thinking, can we have a solution to find out the single-stage solution; that means, let us say auto correlation iterative methods will not be there that I do not have to compute that autocorrelation. So, can I directly compute that α value, or can I directly compute the k value? If I know the k value, I can compute the α value

So, that method is called lattice formulation. So, for part of the next lecture, I will cover that lattice formulation for linear prediction.

Thank you.