UM-SJTU Joint Institute

VV570

Introduction to Engineering Numerical Analysis

Using Linear Predictor to Predict Sound Signals——A Comparison between Two Different Methods

Author
Zhang Congfei

Supervisor
Dr. Jing and Manuel

July 21, 2017

1 ABSTRACT

For speech signals, We can predict the next signal given the previous signals by using the linear speech model. We have two methods to do the prediction, one is the covariance method, and the other is the Autocorrelation method, which can be interpreted as predicting the signal by windowing the error and windowing the signal, respectively. In this paper, we introduce the mathematics basis of the two methods and do experiments to compare these two methods. From the experiments we can see that: covariance methods provides better prediction but larger calculation time, and the autocorrelation method provides prediction with faster calculation time but lower accuracy.

2 Linear prediction of speech

2.1 Covariance Method[1]

Covariance method mininized the error of the given signals to determine the coefficients a_i in the formluar. By the defination of error, we can write E_n by the equation:

$$E_n = \sum_{m=n}^{n+N-1} (y[m] - \sum_{i=1}^{P} a_i y[m-i])^2$$

simplify the form, we can get:

$$E_n = \sum_{m=n}^{n+N-1} \sum_{i=0}^{P} \sum_{j=0}^{P} a_i y[m-i] y[m-j] a_j$$

define $\Phi_n(j,i) = \sum_m^{n+N-1} y[m-i]y[m-j]$
then E_n can be rewrite as

$$E_n = \sum_{i=0}^{P} \sum_{j=0}^{P} a_i \Phi(j,i) a_j$$

from this equation, we want to find the minimum value of error, thus we take the partial derivation of E_n with respect to a_k:

$$\frac{\partial E_n}{\partial k} = 0 = 2 \sum_{i=0}^{P} a_i \Phi_n(k,i)$$

by the above equation we can get:

$$\sum_{m=-\infty}^{\infty} y[m]y[m-k] = \sum_{i=1}^{p} a_i \sum_{m=-\infty}^{\infty} y[m-i]y[m-k]$$

we take $a_0 = 1$ in this equation, thus we can get:

$$\Phi(k,0) = \sum_{i=1}^{P} a_i \Phi_n(k,i) \qquad (1)$$

using equation (1), we can have P equations with P unknowns, write it in linear algebra form, we can change the problem of solving groups of linear equations into solve the matrix:
thus to find the values of a_i we need to prove the existance of the inverse of Φ and find it.

$$\begin{pmatrix} \Phi(1,0) \\ \vdots \\ \Phi(p,0) \end{pmatrix} = \underbrace{\begin{pmatrix} \Phi(1,1) & \cdots & \Phi(1,p) \\ \vdots & & \vdots \\ \Phi(p,1) & \cdots & \Phi(p,p) \end{pmatrix}}_{\Phi} \begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}$$

2.1.1 Finding the inverse of Φ

To solve for the set of equations, we want to use Cholesky decomposition. It is a decomposition of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. e.g.

$$A = L * L^T$$

Every positive definite symmetric matrix can be decomposed has a unique Cholesky decomposition[2] Φ is already symmertic, so we only need to prove Φ is positive definite.
so we need to show that all the eiegn values of Φ is positive.

2.1.2 Prove of existence of unique Cholesky decomposition

We prove it by induction:
for rows columns equal 1, a symmetric matrix is :

$$\Phi_{1,1}$$

its trivial to find its eiegn value is $\phi_{1,1}$,which is a positive value.
assume for rows columns equal to k, the assumption is also true.
For rows = k+1; we need to find

$$det(\Phi_{k+1} - \lambda^{k+1} I) = 0$$

denote λ^k as the set of eiegn values for Φ_k and λ_k as the kth value in λ^k
denote the element in i^{th} row and j^{th} column as $\phi_{i,j}$,
by the following equation:

$$\det \begin{pmatrix} A & B \\ C & D \end{pmatrix} = det(A) * det(D - C * A^{-1} C^T)$$

we can rewrite the equation:

$$det(\Phi_{k+1} - \lambda^{k+1} I) = det(\phi_{1,1} - \lambda_{k+1}) * det(\Phi_{k-1} - \lambda_k I - C * \frac{1}{\phi_{1,1}} * C^{-1})$$

$det(\Phi_{k-1} - \lambda_k I - C * \frac{1}{\phi_{1,1}} * C^{-1})$ is not zero, thus $det(\phi_{1,1} - \lambda_{k+1})$ must be 0,
Thus λ_{k+1} is also positive.
Thus Φ is positive definte.
Proved.

2.1.3 pseudocode of Cholesky decomposition

We write down the psudocode of the algorithm and implment it by matlab.

Cholesky decomposition Algorithm

Input: positive definite matrix A
Output: Upper triangluar matrix L, where $A = L * L^T$

```
len=lenth of A
for i=1:len do
  for m=1:i-1 do
    A(i,i)= A(i,i) - A(m,i)^2
  end for
  A(i,i)=square root of A(i,i)
  for j=i+1:len do
    for m=1:i-1 do
      A(i,j)= A(i,j) - A(m,i)*A(m,j)
    end for
    A(i,j)=A(i,j)/A(i,i)
  end for
end for
B= upper triangular part of A
end algorithm
```

2.2 Autocorrelation Method

With autocorrelation method, we window the whole speech signal. We define the function of autocorrletion as:

$$R(n) = \sum_{m=-\infty}^{\infty} y[m]y[n-m]$$

To perform the calculation, we set all the value outside the interval $n \subset [0, N)$ to be 0.
we can calculate $\Phi(j,k)$ from $-\infty$ to ∞:

$$\Phi(j,k) = \sum_{-\infty}^{\infty} y[n-j]y[n-k]$$

as described before, all the errors would be 0 outside the interval, thus we can rewrite $\Phi(j,k)$ as:

$$\Phi(j,k) = \sum_{0}^{N+p-1} y[n-j]y[n-k]$$

substitute n with $n+j$, we can rewrite the equation as:

$$\Phi(j,k) = \sum_{0}^{N-1-j+k} y[n]y[n+j-k]$$

which equals to $R(j,k)$
thus we can find the equation to find the solution:

$$R(j) = \sum_{k=1}^{p} R(j-k)a_k$$

write it in matrix formluar , we can get:

$$\begin{pmatrix} R(1) \\ R(2) \\ \\ R(p) \end{pmatrix} = \begin{pmatrix} R(0) & R(1) & R(p-1) \\ R(1) & R(0) & R(p-2) \\ \\ R(p-1) & R(p-2) & R(0) \end{pmatrix} * \begin{pmatrix} a_1 \\ a_2 \\ \\ a_p \end{pmatrix}$$

this is called a Toeplitz matrix, with all the elements on diagonal being identical. We will try to solve the euqation using Levinson algorithm.

2.2.1 pseudocode of Levinson algorithm

We write down the psudocode of the algorithm and implment it by matlab.

Levinson algorithm

Input: Toeplitz matrix A,vector b
Output: a_i

r=A[:,0]
$r_0 = r[0]$
$r_{rest} = r[1:]$
y[1]=-r_{rest}[1]
a[1]=b[1]
p=length of b
α=-$r_r est(1)$
β=1
for i=1:p-1 do:
  β=(1-α^2)*β
  ϵ=(b(i+1)-$r_{rest}(1:i)' * fliplr(a(1:i)))/\beta$
  v(1:i)=a(1:i)+ϵ*fliplr(y(1:i))
  a(1:i+1)=[v(1:i);ϵ]
  α=-($r_{rest}(i+1) + r_{rest}(1:i)' * fliplr(y(1:i))\beta$
  z(1:i)=y(1:i)+α*fliplr(y(1:i))
  y(1:i+1)=[z(1:i);α]
end for
a=a/$r_0{}^T$
end algorithm

3 Experiment

By using the wav file 'Track 1.wav', we do the prediction experiment. Using 50 data as sample, we calculate coefficients a_i based on the both covariance method and correlation method. we test the two methods on a single wav file for 100 times over different signal intervals and plot the error of both methods on a single graph. Figure 2.2.1 shows the error plot. We also exam the runtime(in average) of these two methods, shown in table.

Runtime Comparison

	autocorrelation	covariance
Time(s)	0.0019	0.0052

Table 1: Runtime

From figure 1, we can observe that generally, the autocorrelation method provides a larger error than the covariance method. Meanwhile, autocorrelation method runs faster than covariance method. In fact,

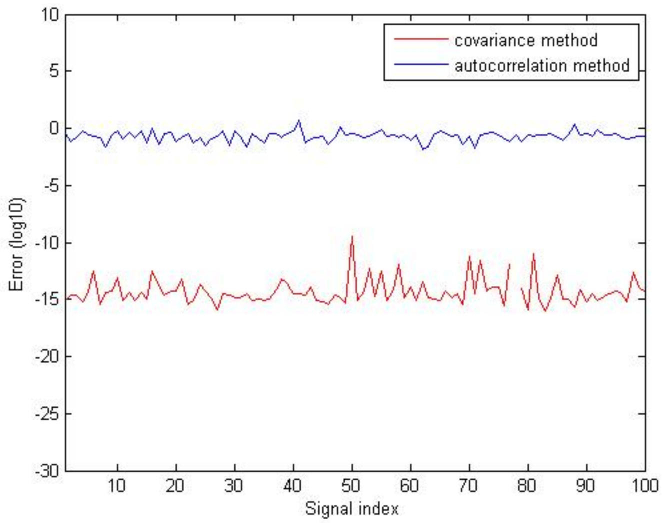

Figure 1: Error

in $O(p^3)$ and $O(p^2)$,respectively[3]. So there is a trade-off in choosing the prediction method. If we require fast prediction had has larger tolerance, we can choose the autocorrelation method. Or if we require high accuracy of prediction, we can switch to covariance method.

4 References

[1] Linear Prediction of Speech, J.D. Markel,A.H. Gray, Jr.

[2] Golub, Van Loan (1996, p. 143), Horn, Johnson (1985, p. 407), Trefethen, Bau (1997, p. 174)

[3] Introduction to Speech Processing, Ricardo Gutierrez-Osuna, CSE@TAMU