

Lecture 26: Linear Prediction

Mark Hasegawa-Johnson

ECE 401: Signal Processing

- ① Linear prediction
- ② Solving for the LPC Coefficients: General Form
- ③ Implementations: Autocovariance and Autocorrelation Methods
- ④ LPC-10 Speech Synthesis Model
- ⑤ Summary

Outline

- 1 Linear prediction
- 2 Solving for the LPC Coefficients: General Form
- 3 Implementations: Autocovariance and Autocorrelation Methods
- 4 LPC-10 Speech Synthesis Model
- 5 Summary

Review: Covariance

- The **covariance** between samples $x[m]$ and $x[m - n]$ is

$$\rho_x[m, m - n] = E [(x[m] - \mu_x[m])(x[m - n] - \mu_x[m - n])^*]$$

- Covariance is interesting because it tells us the best linear prediction of sample $x[m]$ given sample $x[m - n]$:

$$\hat{x}[m] = \mu_x[m] + \frac{\rho_x[m, m - n]}{\sigma_x^2[m - n]} (x[m - n] - \mu_x[m - n])$$

Review: Autocorrelation approximates covariance

- Autocorrelation:

$$\frac{r_x[n]}{N-n} = \frac{1}{N-n} \sum_{m=n}^{N-1} x[m]x^*[m-n] \approx E[x[m]x^*[m-n]]$$

- Covariance:

$$\rho_x[m, m-n] = E[(x[m] - \mu_x[m])(x[m-n] - \mu_x[m-n])^*]$$

- For zero-mean, stationary signals, covariance is the expected average value of autocorrelation

Linear predictive coding (LPC): Synthesis model

A linearly predictable signal is one whose samples, $s[n]$, can be estimated as the linear prediction of p previous samples plus some small error term $e[n]$:

$$s[n] = e[n] + \sum_{k=1}^p a_k s[n - k]$$

LPC prediction model

We can say that $\tilde{s}[n]$ is the predictable part of the signal:

$$\tilde{s}[n] = \sum_{k=1}^p a_k s[n-k]$$

...and the whole signal is the sum of its predictable part plus the error. For this reason the error is sometimes called the excitation:

$$s[n] = \tilde{s}[n] + e[n]$$

The LPC analysis filter

Another way to say the same thing: We model the error, $e[n]$, as being the part of $s[n]$ that cannot be predicted based on a linear combination of the previous p samples:

$$\begin{aligned} e[n] &= s[n] - \tilde{s}[n] \\ &= s[n] - \sum_{k=1}^p a_k s[n-k] \end{aligned}$$

Outline

- 1 Linear prediction
- 2 Solving for the LPC Coefficients: General Form
- 3 Implementations: Autocovariance and Autocorrelation Methods
- 4 LPC-10 Speech Synthesis Model
- 5 Summary

Mean-squared error (MSE)

We will choose a_k to minimize the mean squared error (MSE):

$$\begin{aligned}\mathcal{E} &= \sum_m e^2[m] \\ &= \sum_m \left(s[m] - \sum_{k=1}^p a_k s[m-k] \right)^2\end{aligned}$$

Minimize the MSE

$$\mathcal{E} = \sum_m \left(s[m] - \sum_{k=1}^p a_k s[m-k] \right)^2$$

Differentiate $\mathcal{E}$ with respect to a_i :

$$\frac{\partial \mathcal{E}}{\partial a_i} = 2 \sum_m s[m-i] \left(s[m] - \sum_{k=1}^p a_k s[m-k] \right)$$

... and set the derivative to zero:

$$0 = 2 \sum_m s[m-i] \left(s[m] - \sum_{k=1}^p a_k s[m-k] \right)$$

We get the solution:

$$\sum_m s[m-i] a[m] = \sum_{k=1}^p a_k \sum_m s[m-k] s[m-i]$$

LPC: The Solution!

So we discover that, to minimize $\mathcal{E} = \sum_m e^2[m]$, we should choose the coefficients a_i so that

$$\sum_m s[m-i]s[m] = \sum_{k=1}^p a_k \sum_m s[m-k]s[m-i]$$

Notice that each of those summations is an autocorrelation! The optimum LPC coefficients are given by

$$r_s[i] = \sum_{k=1}^p a_k r_s[k-i]$$

(notice that $r_s[k-i] = r_s[i-k]$)

The LPC solution as a matrix equation

So we discover that, to minimize $\mathcal{E} = \sum_m e^2[m]$, we should choose the coefficients a_i so that

$$r_s[i] = \sum_{k=1}^p a_k r_s[k - i]$$

Now, notice that this is p linear equations (one for each i) in p unknowns (a_1 through a_p), so we can write it as a single matrix equation:

$$\boldsymbol{\gamma} = \mathbf{R}\mathbf{a}$$

$$\boldsymbol{\gamma} = \begin{bmatrix} r_s[1] \\ \vdots \\ r_s[p] \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} r_s[0] & \cdots & r_s[p-1] \\ \vdots & \ddots & \vdots \\ r_s[p-1] & \cdots & r_s[0] \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}$$

The LPC solution as a matrix equation

So we discover that, to minimize $\mathcal{E} = \sum_m e^2[m]$, we only need to invert a matrix!

$$\mathbf{a} = \mathbf{R}^{-1}\boldsymbol{\gamma}$$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} r_s[0] & \cdots & r_s[p-1] \\ \vdots & \ddots & \vdots \\ r_s[p-1] & \cdots & r_s[0] \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} r_s[1] \\ \vdots \\ r_s[p] \end{bmatrix},$$

Try the quiz!

Go to the course web page, and try the quiz!

What is the residual error $\mathcal{E}$?

Now we might ask: What is $\mathcal{E}$ after we've minimized it?

$$\begin{aligned}\mathcal{E} &= \sum_m e^2[m] \\ &= \sum_m \left(s[m] - \sum_{k=1}^p a_k s[m-k] \right)^2 \\ &= r_s[0] - 2 \sum_k a_k r_s[k] + \sum_i \sum_k a_i a_k r_s[k-i]\end{aligned}$$

If we plug in $r_s[k] = \sum_i a_i r_s[k-i]$, it simplifies to

$$\mathcal{E} = r_s[0] - \sum_k a_k r_s[k]$$

What is the residual error $\mathcal{E}$?

Notice that $r_s[0]$ is the energy of the speech signal **before** prediction:

$$r_s[0] = \sum_m s^2[m]$$

So we can say that $\sum_k a_k r_s[k]$ is the portion of the original energy that is explained by linear prediction:

$$\mathcal{E} = r_s[0] - \sum_k a_k r_s[k] = r_s[0] - \mathbf{a}^T \boldsymbol{\gamma}$$
$$\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} r_s[1] \\ \vdots \\ r_s[p] \end{bmatrix}$$

Outline

- 1 Linear prediction
- 2 Solving for the LPC Coefficients: General Form
- 3 Implementations: Autocovariance and Autocorrelation Methods**
- 4 LPC-10 Speech Synthesis Model
- 5 Summary

p linear equations in p unknowns

Remember that the general solution is:

$$\mathbf{a} = \mathbf{R}^{-1}\boldsymbol{\gamma}$$

where $\mathbf{R}$ is a $p \times p$ matrix. Computing its inverse is $\mathcal{O}\{p^3\}$ (I've read that solutions exist that are $\mathcal{O}\{p^{2.3}\}$, but normal Gaussian elimination is $\mathcal{O}\{p^3\}$). We might ask: Is there any way to reduce the computation?

Autocorrelation $\Rightarrow$ Toeplitz matrix

If we window the signal before computing the autocorrelation, then $r_s[i - k]$ only depends on $|i - k|$, so the matrix $\mathbf{R}$ has a special structure called **Toeplitz**:

$$\gamma = \mathbf{R}\mathbf{a}$$

$$\mathbf{R} = \begin{bmatrix} r_s[0] & r_s[1] & \cdots & r_s[p-1] \\ r_s[1] & r_s[0] & \cdots & r_s[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_s[p-1] & r_s[p-2] & \cdots & r_s[0] \end{bmatrix}$$

An equation $\mathbf{a} = \mathbf{R}^{-1}\gamma$ with a Toeplitz $\mathbf{R}$ can be solved in $\mathcal{O}\{p^2\}$ using a special equation called the Levinson-Durbin recursion.

What are the disadvantages of the autocorrelation solution?

The disadvantage of autocorrelation is that we are predicting each sample only using previous samples of the **windowed** speech signal:

$$e[m] = s[m] - \sum_k a_k s[m - k]$$

- The sample at the beginning of the window, $s[0]$, is being predicted based on samples outside the window, which are zero: $s[-p] = s[-1] = 0$!
- Since the error has this built-in artifact, the coefficients will have some built-in error.
- This problem is most important if $w[n]$ is short.

The autocovariance solution: Predict from samples outside the window

- There is another solution, used more often in financial predictions (and less often in signal processing) called the “autocovariance solution.” The autocovariance solution is to predict each sample possibly using samples from before the start time of the window, i.e., use some known nonzero value for $s[-1]$ and $s[-p]$.
- Autocovariance gives more accurate coefficients for short windows, but. . .
- The matrix is no longer Toeplitz (e.g., $\phi_n[1, 1] \neq \phi_n[2, 2]$), so solving $\mathbf{a} = \mathbf{R}^{-1}\boldsymbol{\gamma}$ is an $\mathcal{O}\{p^3\}$ operation.
- More important in real-world applications: autocorrelation gives a synthesis filter that is guaranteed stable, but autocovariance doesn't.
- For those reasons, we'll only use autocorrelation in this class, not autocovariance.

Outline

- 1 Linear prediction
- 2 Solving for the LPC Coefficients: General Form
- 3 Implementations: Autocovariance and Autocorrelation Methods
- 4 LPC-10 Speech Synthesis Model**
- 5 Summary

Speech synthesis using LPC

- LPC-10 was a secure (easily encryptable) speech coding standard used in the 1980s
- It used an impulse train (for voiced sounds) or white noise (for unvoiced sounds) as the input to a 10-tap LPC prediction filter
- You can see the flowchart and hear examples at <https://nmikstas.github.io/portfolio/talkingClock/talkingClock.html>

Outline

- 1 Linear prediction
- 2 Solving for the LPC Coefficients: General Form
- 3 Implementations: Autocovariance and Autocorrelation Methods
- 4 LPC-10 Speech Synthesis Model
- 5 Summary

Summary

- LPC models the resonant structure of speech using a p^{th} -order linear predictor:

$$s[n] = e[n] + \sum_{k=1}^p a_k s[n-k]$$

- Minimizing the mean-squared error results in p linear equations in p unknowns, which can be solved as

$$\boldsymbol{\gamma} = \mathbf{R}\mathbf{a}$$

$$\boldsymbol{\gamma} = \begin{bmatrix} r_s[1] \\ \vdots \\ r_s[p] \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} r_s[0] & \cdots & r_s[p-1] \\ \vdots & \ddots & \vdots \\ r_s[p-1] & \cdots & r_s[0] \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}$$