CS578- Speech Signal Processing

Lecture 7: Speech Coding

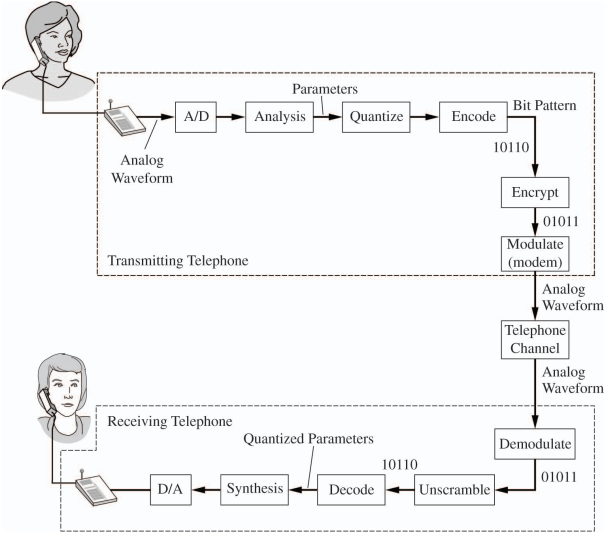
Yannis Stylianou

University of Crete, Computer Science Dept., Multimedia Informatics Lab
yannis@csd.uoc.gr

Univ. of Crete

Outline

Digital telephone communication system



Categories of speech coders

- Waveform coders (16-64 kbps, $f_s = 8000Hz$)
- Hybrid coders (2.4-16 kbps, $f_s = 8000Hz$)
- Vocoders (1.2-4.8 kbps, $f_s = 8000Hz$)

Categories of speech coders

- Waveform coders (16-64 kbps, $f_s = 8000Hz$)
- Hybrid coders (2.4-16 kbps, $f_s = 8000Hz$)
- Vocoders (1.2-4.8 kbps, $f_s = 8000Hz$)

Categories of speech coders

- Waveform coders (16-64 kbps, $f_s = 8000Hz$)
- Hybrid coders (2.4-16 kbps, $f_s = 8000Hz$)
- Vocoders (1.2-4.8 kbps, $f_s = 8000Hz$)

Speech quality

- Closeness of the processed speech waveform to the original speech waveform
- Naturalness
- Background artifacts
- Intelligibility
- Speaker identifiability

Speech quality

- Closeness of the processed speech waveform to the original speech waveform
- Naturalness
- Background artifacts
- Intelligibility
- Speaker identifiability

Speech quality

- Closeness of the processed speech waveform to the original speech waveform
- Naturalness
- Background artifacts
- Intelligibility
- Speaker identifiability

Speech quality

- Closeness of the processed speech waveform to the original speech waveform
- Naturalness
- Background artifacts
- Intelligibility
- Speaker identifiability

Speech quality

- Closeness of the processed speech waveform to the original speech waveform
- Naturalness
- Background artifacts
- Intelligibility
- Speaker identifiability

Measuring Speech Quality

- ▷ Subjective tests:
  - Diagnostic Rhyme Test (DRT)
  - Diagnostic Acceptability Measure (DAM)
  - Mean Opinion Score (MOS)
- ▷ Objective tests:
  - Segmental Signal-to-Noise Ratio (SNR)
  - Articulation Index

Measuring Speech Quality

- Subjective tests:
  - Diagnostic Rhyme Test (DRT)
  - Diagnostic Acceptability Measure (DAM)
  - Mean Opinion Score (MOS)
- Objective tests:
  - Segmental Signal-to-Noise Ratio (SNR)
  - Articulation Index

Measuring Speech Quality

- ▷ Subjective tests:
  - Diagnostic Rhyme Test (DRT)
  - Diagnostic Acceptability Measure (DAM)
  - Mean Opinion Score (MOS)
- ▷ Objective tests:
  - Segmental Signal-to-Noise Ratio (SNR)
  - Articulation Index

Measuring Speech Quality

- ▷ Subjective tests:
  - Diagnostic Rhyme Test (DRT)
  - Diagnostic Acceptability Measure (DAM)
  - Mean Opinion Score (MOS)
- ▷ Objective tests:
  - Segmental Signal-to-Noise Ratio (SNR)
  - Articulation Index

Measuring Speech Quality

- ▷ Subjective tests:
  - Diagnostic Rhyme Test (DRT)
  - Diagnostic Acceptability Measure (DAM)
  - Mean Opinion Score (MOS)
- ▷ Objective tests:
  - Segmental Signal-to-Noise Ratio (SNR)
  - Articulation Index

- Statistical models of speech (preliminary)
- Scalar quantization (i.e., waveform coding)
- Vector quantization (i.e., subband and sinusoidal coding)

Quantization

- Statistical models of speech (preliminary)
- Scalar quantization (i.e., waveform coding)
- Vector quantization (i.e., subband and sinusoidal coding)

- Statistical models of speech (preliminary)
- Scalar quantization (i.e., waveform coding)
- Vector quantization (i.e., subband and sinusoidal coding)

- Classic LPC
- Mixed Excitation Linear Prediction, MELP
- Multipulse LPC
- Code Excited Linear Prediction (CELP)

Linear Prediction Coding, LPC

- Classic LPC
- Mixed Excitation Linear Prediction, MELP
- Multipulse LPC
- Code Excited Linear Prediction (CELP)

Linear Prediction Coding, LPC

- Classic LPC
- Mixed Excitation Linear Prediction, MELP
- Multipulse LPC
- Code Excited Linear Prediction (CELP)

Linear Prediction Coding, LPC

- Classic LPC
- Mixed Excitation Linear Prediction, MELP
- Multipulse LPC
- Code Excited Linear Prediction (CELP)

Outline

Probability Density of speech

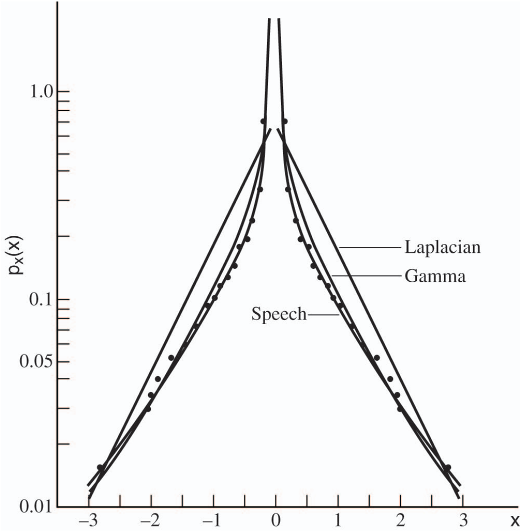
By setting $x[n] \rightarrow x$, the histogram of speech samples can be approximated by a *gamma density*:

$$p_X(x) = \left(\frac{\sqrt{3}}{8\pi\sigma_x |x|} \right)^{1/2} e^{-\frac{\sqrt{3}|x|}{2\sigma_x}}$$

or by a simpler *Laplacian density*:

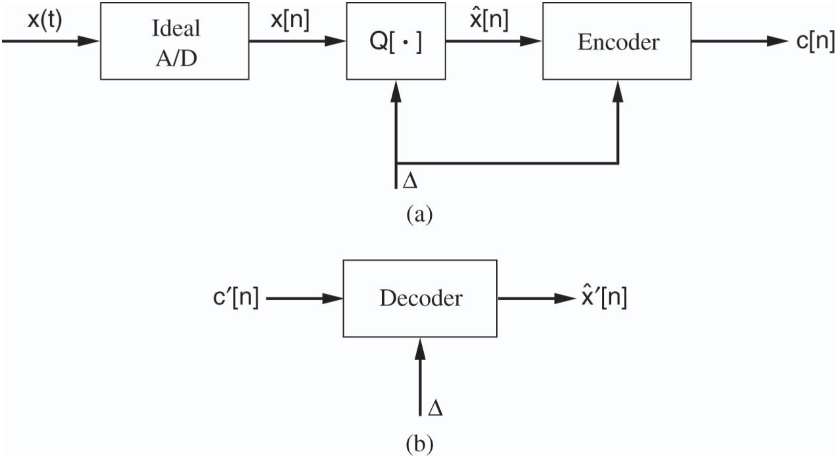
$$p_X(x) = \frac{1}{\sqrt{2}\sigma_x} e^{-\frac{\sqrt{3}|x|}{\sigma_x}}$$

Densities comparison



Outline

Coding and Decoding



Fundamentals of Scalar Coding

- Let's quantize a single sample speech value, $x[n]$ into M *reconstruction* or *decision* levels:

$$\hat{x}[n] = \hat{x}_i = Q(x[n]), \quad x_{i-1} < x[n] \leq x_i$$

  with $1 \leq i \leq M$ and x_k denotes the M decision levels with $0 \leq k \leq M$.
- Assign a *codeword* in each reconstruction level. Collection of codewords makes a *codebook*.
- Using B-bit binary codebook we can represent each 2^B different *quantization* (reconstruction) levels.
- *Bit rate*, I, is defined as: $I = Bf_s$

- Let's quantize a single sample speech value, $x[n]$ into M *reconstruction* or *decision* levels:

$$\hat{x}[n] = \hat{x}_i = Q(x[n]), \quad x_{i-1} < x[n] \leq x_i$$

with $1 \leq i \leq M$ and x_k denotes the M decision levels with $0 \leq k \leq M$.

- Assign a *codeword* in each reconstruction level. Collection of codewords makes a *codebook*.
- Using B-bit binary codebook we can represent each 2^B different *quantization* (reconstruction) levels.
- *Bit rate*, I, is defined as: $I = Bf_s$

Fundamentals of Scalar Coding

- Let's quantize a single sample speech value, $x[n]$ into M *reconstruction* or *decision* levels:

$$\hat{x}[n] = \hat{x}_i = Q(x[n]), \quad x_{i-1} < x[n] \leq x_i$$

  with $1 \leq i \leq M$ and x_k denotes the M decision levels with $0 \leq k \leq M$.

- Assign a *codeword* in each reconstruction level. Collection of codewords makes a *codebook*.
- Using B-bit binary codebook we can represent each 2^B different *quantization* (reconstruction) levels.
- *Bit rate*, I, is defined as: $I = Bf_s$

Fundamentals of Scalar Coding

- Let's quantize a single sample speech value, $x[n]$ into M *reconstruction* or *decision* levels:

$$\hat{x}[n] = \hat{x}_i = Q(x[n]), \quad x_{i-1} < x[n] \leq x_i$$

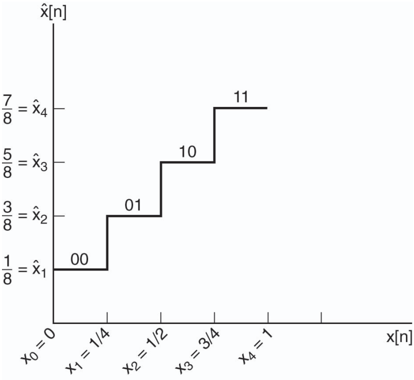
  with $1 \leq i \leq M$ and x_k denotes the M decision levels with $0 \leq k \leq M$.
- Assign a *codeword* in each reconstruction level. Collection of codewords makes a *codebook*.
- Using B-bit binary codebook we can represent each 2^B different *quantization* (reconstruction) levels.
- *Bit rate*, I, is defined as: $I = Bf_s$

Uniform quantization

$$x_i - x_{i-1} = \Delta, \quad 1 \leq i \leq M$$
$$\hat{x}_i = \frac{x_i + x_{i-1}}{2}, \quad 1 \leq i \leq M$$

Δ is referred to as uniform quantization step size.

▷ Example of a 2-bit *uniform quantization*:



Uniform quantization: designing decision regions

- Signal range: $-4\sigma_x \leq x[n] \leq 4\sigma_x$
- Assuming B-bit binary codebook, we get 2^B quantization (reconstruction) levels
- Quantization step size, Δ:

$$\Delta = \frac{2x_{max}}{2^B}$$

- Δ and *quantization noise*.

Classes of Quantization noise

There are two classes of quantization noise:

- Granular Distortion:

$$\hat{x}[n] = x[n] + e[n]$$

  where $e[n]$ is the quantization noise, with:

$$-\frac{\Delta}{2} \leq e[n] \leq \frac{\Delta}{2}$$

- Overload Distortion: *clipped samples*

Assumptions

- Quantization noise is an ergodic white-noise random process:

$$\begin{aligned} r_e[m] &= E(e[n]e[n+m]) \\ &= \sigma_e^2, \quad m = 0 \\ &= 0, \quad m \neq 0 \end{aligned}$$

- Quantization noise and input signal are uncorrelated:

$$E(x[n]e[n+m]) = 0 \quad \forall m$$

- Quantization noise is uniform over the quantization interval

$$\begin{aligned} p_e(e) &= \frac{1}{\Delta}, \quad -\frac{\Delta}{2} \leq e \leq \frac{\Delta}{2} \\ &= 0, \quad \text{otherwise} \end{aligned}$$

Assumptions

- Quantization noise is an ergodic white-noise random process:

$$\begin{aligned} r_e[m] &= E(e[n]e[n+m]) \\ &= \sigma_e^2, \quad m = 0 \\ &= 0, \quad m \neq 0 \end{aligned}$$

- Quantization noise and input signal are uncorrelated:

$$E(x[n]e[n+m]) = 0 \;\; \forall m$$

- Quantization noise is uniform over the quantization interval

$$\begin{aligned} p_e(e) &= \frac{1}{\Delta}, \quad -\frac{\Delta}{2} \leq e \leq \frac{\Delta}{2} \\ &= 0, \quad \text{otherwise} \end{aligned}$$

Assumptions

- Quantization noise is an ergodic white-noise random process:

$$\begin{aligned} r_e[m] &= E(e[n]e[n+m]) \\ &= \sigma_e^2, \quad m = 0 \\ &= 0, \quad m \neq 0 \end{aligned}$$

- Quantization noise and input signal are uncorrelated:

$$E(x[n]e[n+m]) = 0 \ \ \forall m$$

- Quantization noise is uniform over the quantization interval

$$\begin{aligned} p_e(e) &= \tfrac{1}{\Delta}, \quad -\tfrac{\Delta}{2} \leq e \leq \tfrac{\Delta}{2} \\ &= 0, \quad \text{otherwise} \end{aligned}$$

Dithering

Definition (Dithering)

We can force $e[n]$ to be white and uncorrelated with $x[n]$ by adding noise to $x[n]$ before quantization!

Signal-to-Noise Ratio

- To quantify the severity of the quantization noise, we define the *Signal-to-Noise Ratio* (SNR) as:

$$\begin{aligned} SNR &= \frac{\sigma_x^2}{\sigma_e^2} \\ &= \frac{E(x^2[n])}{E(e^2[n])} \\ &\approx \frac{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}{\frac{1}{N}\sum_{n=0}^{N-1} e^2[n]} \end{aligned}$$

- For uniform pdf and quantizer range $2x_{max}$:

$$\begin{aligned} \sigma_e^2 &= \frac{\Delta^2}{12} \\ &= \frac{x_{max}^2}{3\ 2^{2B}} \end{aligned}$$

- Or

$$SNR = \frac{3\ 2^{2B}}{(\frac{x_{max}}{\sigma_x})^2}$$

- and in dB:

$$SNR(dB) \approx 6B + 4.77 - 20\log_{10}\left(\frac{x_{max}}{\sigma_x}\right)$$

- and since $x_{max} = 4\sigma_x$:

$$SNR(dB) \approx 6B - 7.2$$

Signal-to-Noise Ratio

- To quantify the severity of the quantization noise, we define the *Signal-to-Noise Ratio* (SNR) as:

$$
\begin{aligned}
SNR &= \frac{\sigma_x^2}{\sigma_e^2} \\
&= \frac{E(x^2[n])}{E(e^2[n])} \\
&\approx \frac{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}{\frac{1}{N}\sum_{n=0}^{N-1} e^2[n]}
\end{aligned}
$$

- For uniform pdf and quantizer range $2x_{max}$:

$$
\begin{aligned}
\sigma_e^2 &= \frac{\Delta^2}{12} \\
&= \frac{x_{max}^2}{3\ 2^{2B}}
\end{aligned}
$$

- Or

$$SNR = \frac{3\ 2^{2B}}{(\frac{x_{max}}{\sigma_x})^2}$$

- and in dB:

$$SNR(dB) \approx 6B + 4.77 - 20\log_{10}\left(\frac{x_{max}}{\sigma_x}\right)$$

- and since $x_{max} = 4\sigma_x$:

$$SNR(dB) \approx 6B - 7.2$$

Signal-to-Noise Ratio

- To quantify the severity of the quantization noise, we define the *Signal-to-Noise Ratio* (SNR) as:

$$\begin{aligned} SNR &= \frac{\sigma_x^2}{\sigma_e^2} \\ &= \frac{E(x^2[n])}{E(e^2[n])} \\ &\approx \frac{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}{\frac{1}{N}\sum_{n=0}^{N-1} e^2[n]} \end{aligned}$$

- For uniform pdf and quantizer range $2x_{max}$:

$$\begin{aligned} \sigma_e^2 &= \frac{\Delta^2}{12} \\ &= \frac{x_{max}^2}{3\ 2^{2B}} \end{aligned}$$

- Or

$$SNR = \frac{3\ 2^{2B}}{(\frac{x_{max}}{\sigma_x})^2}$$

- and in dB:

$$SNR(dB) \approx 6B + 4.77 - 20\log_{10}\left(\frac{x_{max}}{\sigma_x}\right)$$

- and since $x_{max} = 4\sigma_x$:

$$SNR(dB) \approx 6B - 7.2$$

Signal-to-Noise Ratio

- To quantify the severity of the quantization noise, we define the *Signal-to-Noise Ratio* (SNR) as:

$$
\begin{aligned}
SNR &= \frac{\sigma_x^2}{\sigma_e^2} \\
&= \frac{E(x^2[n])}{E(e^2[n])} \\
&\approx \frac{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}{\frac{1}{N}\sum_{n=0}^{N-1} e^2[n]}
\end{aligned}
$$

- For uniform pdf and quantizer range $2x_{max}$:

$$
\begin{aligned}
\sigma_e^2 &= \frac{\Delta^2}{12} \\
&= \frac{x_{max}^2}{3\ 2^{2B}}
\end{aligned}
$$

- Or

$$SNR = \frac{3\ 2^{2B}}{(\frac{x_{max}}{\sigma_x})^2}$$

- and in dB:

$$SNR(dB) \approx 6B + 4.77 - 20\log_{10}\left(\frac{x_{max}}{\sigma_x}\right)$$

- and since $x_{max} = 4\sigma_x$:

$$SNR(dB) \approx 6B - 7.2$$

Signal-to-Noise Ratio

- To quantify the severity of the quantization noise, we define the *Signal-to-Noise Ratio* (SNR) as:

$$
\begin{aligned}
SNR &= \frac{\sigma_x^2}{\sigma_e^2} \\
&= \frac{E(x^2[n])}{E(e^2[n])} \\
&\approx \frac{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}{\frac{1}{N}\sum_{n=0}^{N-1} e^2[n]}
\end{aligned}
$$

- For uniform pdf and quantizer range $2x_{max}$:

$$
\begin{aligned}
\sigma_e^2 &= \frac{\Delta^2}{12} \\
&= \frac{x_{max}^2}{3\ 2^{2B}}
\end{aligned}
$$

- Or

$$SNR = \frac{3\ 2^{2B}}{(\frac{x_{max}}{\sigma_x})^2}$$

- and in dB:

$$SNR(dB) \approx 6B + 4.77 - 20\log_{10}\left(\frac{x_{max}}{\sigma_x}\right)$$

- and since $x_{max} = 4\sigma_x$:

$$SNR(dB) \approx 6B - 7.2$$

Pulse Code Modulation, PCM

- B bits of information per sample are transmitted as a codeword
- instantaneous coding
- not signal-specific
- 11 bits are required for "toll quality"
- what is the rate for $f_s = 10kHz$?
- For CD, with $f_s = 44100$ and $B = 16$ (16-bit PCM), what is the SNR?

Pulse Code Modulation, PCM

- B bits of information per sample are transmitted as a codeword
- instantaneous coding
- not signal-specific
- 11 bits are required for "toll quality"
- what is the rate for $f_s = 10kHz$?
- For CD, with $f_s = 44100$ and $B = 16$ (16-bit PCM), what is the SNR?

Pulse Code Modulation, PCM

- B bits of information per sample are transmitted as a codeword
- instantaneous coding
- not signal-specific
- 11 bits are required for "toll quality"
- what is the rate for $f_s = 10kHz$?
- For CD, with $f_s = 44100$ and $B = 16$ (16-bit PCM), what is the SNR?

Pulse Code Modulation, PCM

- B bits of information per sample are transmitted as a codeword
- instantaneous coding
- not signal-specific
- 11 bits are required for "toll quality"
- what is the rate for $f_s = 10kHz$?
- For CD, with $f_s = 44100$ and $B = 16$ (16-bit PCM), what is the SNR?

Pulse Code Modulation, PCM

- B bits of information per sample are transmitted as a codeword
- instantaneous coding
- not signal-specific
- 11 bits are required for "toll quality"
- what is the rate for $f_s = 10kHz$?
- For CD, with $f_s = 44100$ and $B = 16$ (16-bit PCM), what is the SNR?

Pulse Code Modulation, PCM

- B bits of information per sample are transmitted as a codeword
- instantaneous coding
- not signal-specific
- 11 bits are required for "toll quality"
- what is the rate for $f_s = 10kHz$?
- For CD, with $f_s = 44100$ and $B = 16$ (16-bit PCM), what is the SNR?

Optimal decision and reconstruction level

if $x[n] \mapsto p_x(x)$ we determine the optimal decision level, x_i and the reconstruction level, $\hat{x}$, by minimizing:

$$\begin{aligned} D &= E[(\hat{x} - x)^2] \\ &= \int_{-\infty}^{\infty} p_x(x)(\hat{x} - x)^2 dx \end{aligned}$$

and assuming M reconstruction levels $\hat{x} = Q[x]$:

$$D = \sum_{i=1}^{M} = \int_{x_{i-1}}^{x_i} p_x(x)(\hat{x}_i - x)^2 dx$$

So:

$$\begin{aligned} \frac{\partial D}{\partial \hat{x}_k} &= 0, \quad 1 \leq k \leq M \\ \frac{\partial D}{\partial x_k} &= 0, \quad 1 \leq k \leq M-1 \end{aligned}$$

Optimal decision and reconstruction level

if $x[n] \mapsto p_x(x)$ we determine the optimal decision level, x_i and the reconstruction level, $\hat{x}$, by minimizing:

$$\begin{aligned} D &= E[(\hat{x}-x)^2] \\ &= \int_{-\infty}^{\infty} p_x(x)(\hat{x}-x)^2 dx \end{aligned}$$

and assuming M reconstruction levels $\hat{x} = Q[x]$:

$$D = \sum_{i=1}^{M} = \int_{x_{i-1}}^{x_i} p_x(x)(\hat{x}_i - x)^2 dx$$

So:

$$\begin{aligned} \frac{\partial D}{\partial \hat{x}_k} &= 0, \quad 1 \le k \le M \\ \frac{\partial D}{\partial x_k} &= 0, \quad 1 \le k \le M-1 \end{aligned}$$

Optimal decision and reconstruction level

if $x[n] \mapsto p_x(x)$ we determine the optimal decision level, x_i and the reconstruction level, $\hat{x}$, by minimizing:

$$\begin{aligned} D &= E[(\hat{x}-x)^2] \\ &= \int_{-\infty}^{\infty} p_x(x)(\hat{x}-x)^2 dx \end{aligned}$$

and assuming M reconstruction levels $\hat{x} = Q[x]$:

$$D = \sum_{i=1}^{M} = \int_{x_{i-1}}^{x_i} p_x(x)(\hat{x}_i - x)^2 dx$$

So:

$$\begin{aligned} \frac{\partial D}{\partial \hat{x}_k} &= 0, \quad 1 \leq k \leq M \\ \frac{\partial D}{\partial x_k} &= 0, \quad 1 \leq k \leq M-1 \end{aligned}$$

Optimal decision and reconstruction level, *cont.*

- The minimization of D over decision level, x_k, gives:

$$x_k = \frac{\hat{x}_{k+1} + \hat{x}_k}{2}, \quad 1 \leq k \leq M - 1$$

- The minimization of D over reconstruction level, $\hat{x}_k$, gives:

$$\begin{aligned} \hat{x}_k &= \int_{x_{k-1}}^{x_k} \left[\frac{p_x(x)}{\int_{x_{k-1}}^{x_k} p_x(s)ds} \right] x dx \\ &= \int_{x_{k-1}}^{x_k} \tilde{p}_x(x) x dx \end{aligned}$$

Optimal decision and reconstruction level, *cont.*

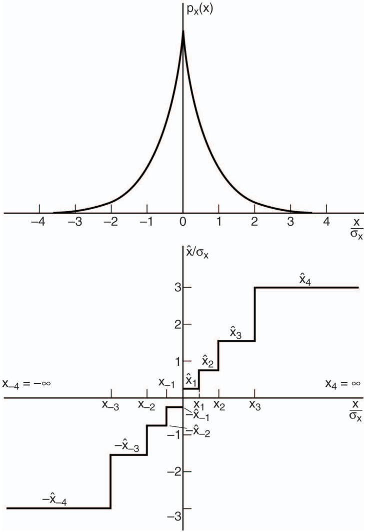
- The minimization of D over decision level, x_k, gives:

$$x_k = \frac{\hat{x}_{k+1} + \hat{x}_k}{2}, \quad 1 \leq k \leq M-1$$

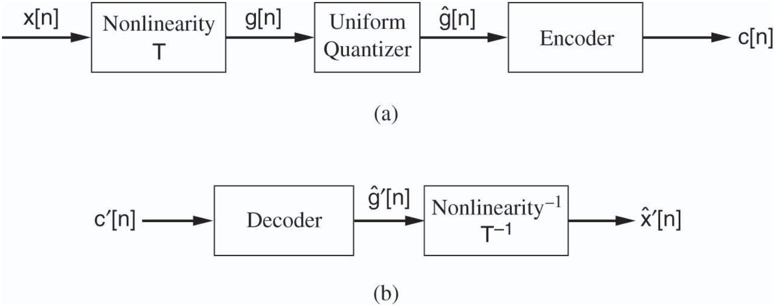
- The minimization of D over reconstruction level, $\hat{x}_k$, gives:

$$\begin{aligned} \hat{x}_k &= \int_{x_{k-1}}^{x_k} \left[\frac{p_x(x)}{\int_{x_{k-1}}^{x_k} p_x(s)ds} \right] x dx \\ &= \int_{x_{k-1}}^{x_k} \tilde{p}_x(x) x dx \end{aligned}$$

Example with Laplacian pdf



Principle of Companding



Companding examples:

- Transformation to a uniform density:

$$
\begin{aligned}
g[n] = T(x[n]) &= \int_{-\infty}^{x[n]} p_x(s)ds - \tfrac{1}{2}, \quad \tfrac{-1}{2} \leq g[n] \leq \tfrac{1}{2} \\
&= 0 \qquad \text{elsewhere}
\end{aligned}
$$

- μ-law:

$$T(x[n]) = x_{max} \frac{\log\left(1 + \mu \frac{|x[n]|}{x_{max}}\right)}{\log\left(1+\mu\right)} sign(x[n])$$

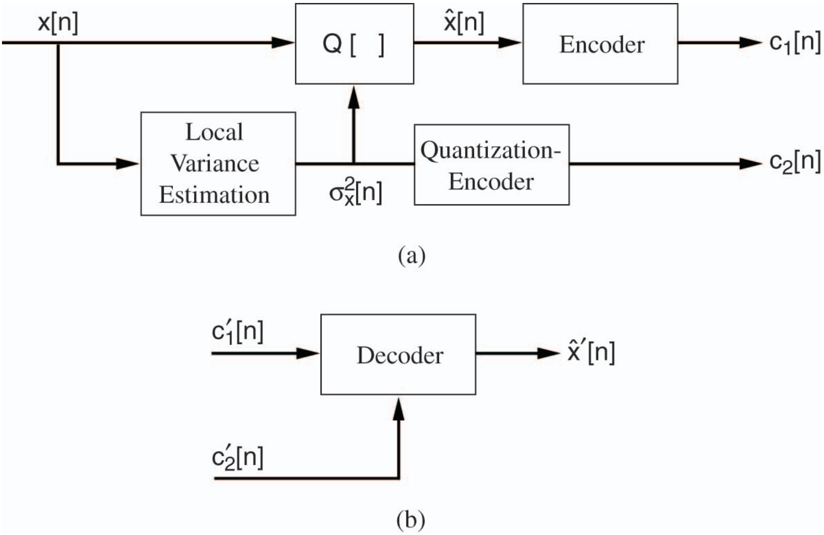
Companding examples:

- Transformation to a uniform density:

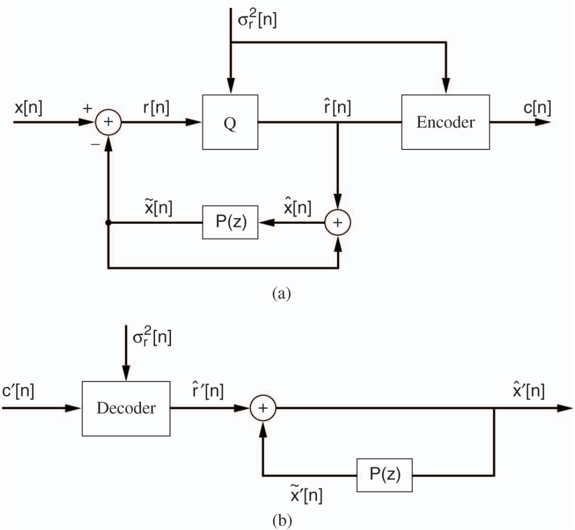
$$\begin{array}{rcl} g[n] = T(x[n]) & = & \int_{-\infty}^{x[n]} p_x(s)ds - \frac{1}{2}, \quad \frac{-1}{2} \leq g[n] \leq \frac{1}{2} \\ & = & 0 \qquad\qquad \text{elsewhere} \end{array}$$

- μ-law:

$$T(x[n]) = x_{max} \frac{\log\left(1 + \mu \frac{|x[n]|}{x_{max}}\right)}{\log\left(1 + \mu\right)} sign(x[n])$$

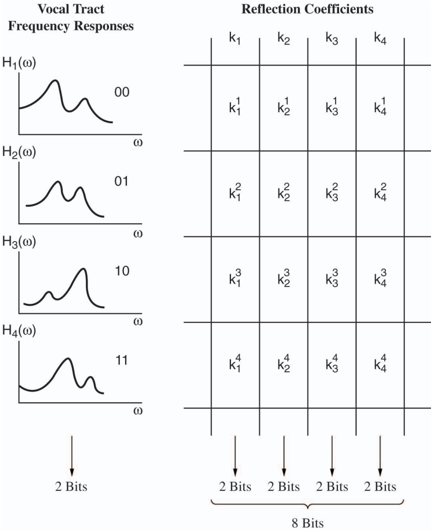

x[n]
Q[ ]
x̂[n]
Encoder
c1[n]
Local Variance Estimation
σx2[n]
Quantization-Encoder
c2[n]
(a)
c'1[n]
Decoder
x̂'[n]
c'2[n]
(b)

Differential and Residual quantization

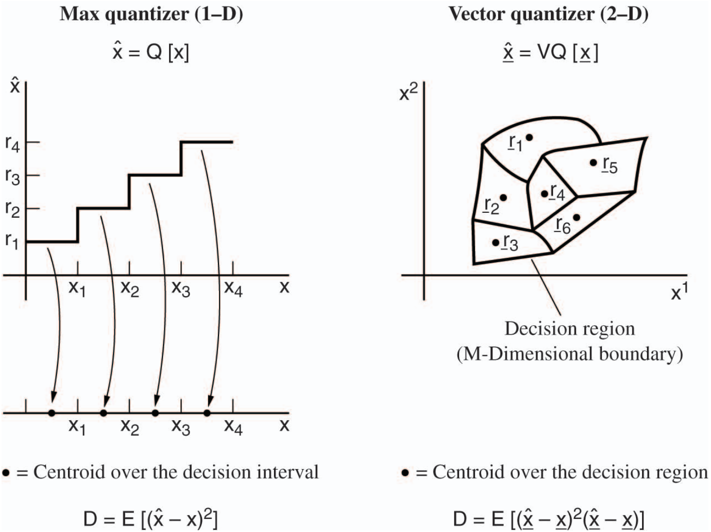


Outline

Motivation for VQ



Comparing scalar and vector quantization



Distortion in VQ

Here we have a multidimensional pdf $p_{\mathbf{x}}(\mathbf{x})$:

$$
\begin{aligned}
D &= E[(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})] \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(\hat{\mathbf{x}} - x)^T(\hat{\mathbf{x}} - \mathbf{x})p_{\mathbf{x}}(\mathbf{x})d\mathbf{x} \\
&= \sum_{i=1}^{M}\int\int_{\mathbf{x}\in\mathcal{C}_i}\cdots\int(\mathbf{r}_i - \mathbf{x})^T(\mathbf{r}_i - \mathbf{x})p_{\mathbf{x}}(\mathbf{x})d\mathbf{x}
\end{aligned}
$$

Two constraints:

- A vector $\mathbf{x}$ must be quantized to a reconstruction level $\mathbf{r}_i$ that gives the smallest distortion:

$$\mathcal{C}_i = \{\mathbf{x} : ||\mathbf{x} - \mathbf{r}_i||^2 \leq ||\mathbf{x} - \mathbf{r}_l||^2, \forall l = 1, 2, \cdots, M\}$$

- Each reconstruction level $\mathbf{r}_i$ must be the centroid of the corresponding decision region, i.e., of the cell $\mathcal{C}_i$:

$$\mathbf{r}_i = \frac{\sum_{\mathbf{x}_m\in\mathcal{C}_i}\mathbf{x}_m}{\sum_{\mathbf{x}_m\in\mathcal{C}_i}1} \quad i = 1, 2, \cdots, M$$

Distortion in VQ

Here we have a multidimensional pdf $p_{\mathbf{x}}(\mathbf{x})$:

$$\begin{aligned} D &= E[(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(\hat{\mathbf{x}} - x)^T(\hat{\mathbf{x}} - \mathbf{x})p_{\mathbf{x}}(\mathbf{x})d\mathbf{x} \\ &= \sum_{i=1}^{M}\int\int_{\mathbf{x}\in\mathcal{C}_i}\cdots\int(\mathbf{r}_i - \mathbf{x})^T(\mathbf{r}_i - \mathbf{x})p_{\mathbf{x}}(\mathbf{x})d\mathbf{x} \end{aligned}$$

Two constraints:

- A vector $\mathbf{x}$ must be quantized to a reconstruction level $\mathbf{r}_i$ that gives the smallest distortion:

$$\mathcal{C}_i = \{\mathbf{x} : ||\mathbf{x} - \mathbf{r}_i||^2 \leq ||\mathbf{x} - \mathbf{r}_l||^2, \forall l = 1, 2, \cdots, M\}$$

- Each reconstruction level $\mathbf{r}_i$ must be the centroid of the corresponding decision region, i.e., of the cell $\mathcal{C}_i$:

$$\mathbf{r}_i = \frac{\sum_{\mathbf{x}_m\in\mathcal{C}_i}\mathbf{x}_m}{\sum_{\mathbf{x}_m\in\mathcal{C}_i}1} \quad i = 1, 2, \cdots, M$$

The k-means algorithm

- S1:

$$D = \frac{1}{N}\sum_{k=0}^{N-1}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T(\hat{\mathbf{x}}_k - \mathbf{x}_k)$$

- S2: Pick an initial guess at the reconstruction levels $\{\mathbf{r}_i\}$
- S3: For each $\mathbf{x}_k$ elect an $\mathbf{r}_i$ closest to $\mathbf{x}_k$. Form clusters (*clustering step*)
- S4: Find the mean of $\mathbf{x}_k$ in each cluster which gives a new $\mathbf{r}_i$. Compute D.
- S5: Stop when the change in D over two consecutive iterations is insignificant.

The LBG algorithm

- Set the *desired* number of cells: $M = 2^B$
- Set an initial codebook $\mathcal{C}^{(0)}$ with *one* codevector which is set as the average of the entire training sequence, $\mathbf{x}_k, \quad k = 1, 2, \cdots, N$.
- Split the codevector into two and get an *initial* new codebook $\mathcal{C}^{(1)}$.
- Perform a k-means algorithm to optimize the codebook and get the *final* $\mathcal{C}^{(1)}$
- Split the final codevectors into four and repeat the above process until the desired number of cells is reached.

The LBG algorithm

- Set the *desired* number of cells: $M = 2^B$
- Set an initial codebook $\mathcal{C}^{(0)}$ with *one* codevector which is set as the average of the entire training sequence, $\mathbf{x}_k, \quad k = 1, 2, \cdots, N$.
- Split the codevector into two and get an *initial* new codebook $\mathcal{C}^{(1)}$.
- Perform a k-means algorithm to optimize the codebook and get the *final* $\mathcal{C}^{(1)}$
- Split the final codevectors into four and repeat the above process until the desired number of cells is reached.

The LBG algorithm

- Set the *desired* number of cells: $M = 2^B$
- Set an initial codebook $\mathcal{C}^{(0)}$ with *one* codevector which is set as the average of the entire training sequence, $\mathbf{x}_k, \quad k = 1, 2, \cdots, N$.
- Split the codevector into two and get an *initial* new codebook $\mathcal{C}^{(1)}$.
- Perform a k-means algorithm to optimize the codebook and get the *final* $\mathcal{C}^{(1)}$
- Split the final codevectors into four and repeat the above process until the desired number of cells is reached.

The LBG algorithm

- Set the *desired* number of cells: $M = 2^B$
- Set an initial codebook $\mathcal{C}^{(0)}$ with *one* codevector which is set as the average of the entire training sequence,
  $\mathbf{x}_k, \quad k = 1, 2, \cdots, N.$
- Split the codevector into two and get an *initial* new codebook $\mathcal{C}^{(1)}$.
- Perform a k-means algorithm to optimize the codebook and get the *final* $\mathcal{C}^{(1)}$
- Split the final codevectors into four and repeat the above process until the desired number of cells is reached.

The LBG algorithm

- Set the *desired* number of cells: $M = 2^B$
- Set an initial codebook $\mathcal{C}^{(0)}$ with *one* codevector which is set as the average of the entire training sequence, $\mathbf{x}_k, \quad k = 1, 2, \cdots, N$.
- Split the codevector into two and get an *initial* new codebook $\mathcal{C}^{(1)}$.
- Perform a k-means algorithm to optimize the codebook and get the *final* $\mathcal{C}^{(1)}$
- Split the final codevectors into four and repeat the above process until the desired number of cells is reached.

Outline

Basic coding scheme in LPC

- Vocal tract system function:

$$H(z) = \frac{A}{1 - P(z)}$$

where

$$P(z) = \sum_{k=1}^{p} a_k z^{-1}$$

- Input is binary: impulse/noise excitation.
- If frame rate is 100 frames/s and we use 13 parameters ($p = 10$, 1 for Gain, 1 for pitch, 1 for voicing decision) we need 1300 parameters/s, instead of 8000 samples/s for $f_s = 8000 Hz$.

Basic coding scheme in LPC

- Vocal tract system function:

$$H(z) = \frac{A}{1 - P(z)}$$

where

$$P(z) = \sum_{k=1}^{p} a_k z^{-1}$$

- Input is binary: impulse/noise excitation.
- If frame rate is 100 frames/s and we use 13 parameters ($p = 10$, 1 for Gain, 1 for pitch, 1 for voicing decision) we need 1300 parameters/s, instead of 8000 samples/s for $f_s = 8000Hz$.

Basic coding scheme in LPC

- Vocal tract system function:

$$H(z) = \frac{A}{1 - P(z)}$$

where

$$P(z) = \sum_{k=1}^{p} a_k z^{-1}$$

- Input is binary: impulse/noise excitation.
- If frame rate is 100 frames/s and we use 13 parameters ($p = 10$, 1 for Gain, 1 for pitch, 1 for voicing decision) we need 1300 parameters/s, instead of 8000 samples/s for $f_s = 8000 Hz$.

Scalar quantization within LPC

For 7200 bps:

- Voiced/unvoiced decision: 1 bit
- Pitch (if voiced): 6 bits (uniform)
- Gain: 5 bits (nonuniform)
- Poles d_i: 10 bits (nonuniform) [5 bits for frequency and 5 bits for bandwidth] $\times$ 6 poles $=$ 60 bits

So: $(1 + 6 + 5 + 60) \times 100$ frames/s $=$ 7200 bps

Scalar quantization within LPC

For 7200 bps:

- Voiced/unvoiced decision: 1 bit
- Pitch (if voiced): 6 bits (uniform)
- Gain: 5 bits (nonuniform)
- Poles d_i: 10 bits (nonuniform) [5 bits for frequency and 5 bits for bandwidth] $\times$ 6 poles $=$ 60 bits

So: $(1 + 6 + 5 + 60) \times 100$ frames/s $=$ 7200 bps

Refinements to the basic LPC coding scheme

- Companding in the form of a logarithmic operator on pitch and gain
- Instead of poles use the reflection (or the PARCOR) coefficients k_i,(nonuniform)
- Companding of k_i:

$$\begin{aligned} g_i &= T[k_i] \\ &= \log\left(\frac{1-k_i}{1+k_i}\right) \end{aligned}$$

- Coefficients g_i can be coded at 5-6 bits each! (which results in 4800 bps for an order 6 predictor, and 100 frames/s)
- Reduce the frame rate by a factor of two (50 frames/s) gives us a bit rate of 2400 bps

REFINEMENTS TO THE BASIC LPC CODING SCHEME

- Companding in the form of a logarithmic operator on pitch and gain
- Instead of poles use the reflection (or the PARCOR) coefficients k_i,(nonuniform)
- Companding of k_i:

$$\begin{aligned} g_i &= T[k_i] \\ &= \log\left(\frac{1-k_i}{1+k_i}\right) \end{aligned}$$

- Coefficients g_i can be coded at 5-6 bits each! (which results in 4800 bps for an order 6 predictor, and 100 frames/s)
- Reduce the frame rate by a factor of two (50 frames/s) gives us a bit rate of 2400 bps

Refinements to the basic LPC coding scheme

- Companding in the form of a logarithmic operator on pitch and gain
- Instead of poles use the reflection (or the PARCOR) coefficients k_i,(nonuniform)
- Companding of k_i:

$$\begin{aligned} g_i &= T[k_i] \\ &= \log\left(\frac{1-k_i}{1+k_i}\right) \end{aligned}$$

- Coefficients g_i can be coded at 5-6 bits each! (which results in 4800 bps for an order 6 predictor, and 100 frames/s)
- Reduce the frame rate by a factor of two (50 frames/s) gives us a bit rate of 2400 bps

Refinements to the basic LPC coding scheme

- Companding in the form of a logarithmic operator on pitch and gain
- Instead of poles use the reflection (or the PARCOR) coefficients k_i,(nonuniform)
- Companding of k_i:

$$\begin{aligned} g_i &= T[k_i] \\ &= \log\left(\frac{1-k_i}{1+k_i}\right) \end{aligned}$$

- Coefficients g_i can be coded at 5-6 bits each! (which results in 4800 bps for an order 6 predictor, and 100 frames/s)
- Reduce the frame rate by a factor of two (50 frames/s) gives us a bit rate of 2400 bps

Refinements to the basic LPC coding scheme

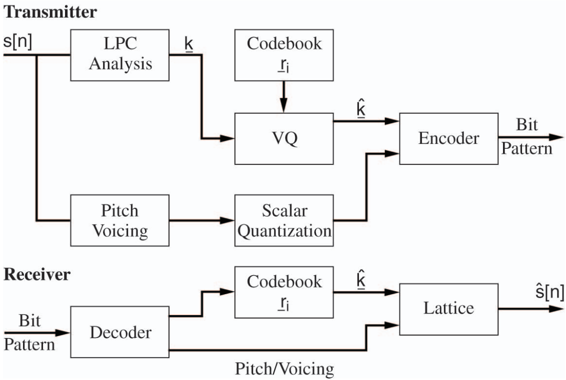
- Companding in the form of a logarithmic operator on pitch and gain
- Instead of poles use the reflection (or the PARCOR) coefficients k_i,(nonuniform)
- Companding of k_i:

$$\begin{aligned} g_i &= T[k_i] \\ &= \log\left(\frac{1-k_i}{1+k_i}\right) \end{aligned}$$

- Coefficients g_i can be coded at 5-6 bits each! (which results in 4800 bps for an order 6 predictor, and 100 frames/s)
- Reduce the frame rate by a factor of two (50 frames/s) gives us a bit rate of 2400 bps

VQ IN LPC CODING

▷ A 10-bit codebook (1024 codewords), 800 bps VQ provides a comparable quality to a 2400 bps scalar quantizer.
▷ A 22-bit codebook (4200000 codewords), 2400 bps VQ provides a higher output speech quality.



Unique components of MELP

- Mixed pulse and noise excitation
- Periodic or aperiodic pulses
- Adaptive spectral enhancements
- Pulse dispersion filter

Line Spectral Frequencies (LSFs) in MELP

LSFs for a pth order all-pole model are defines as follows:

1. Form two polynomials:

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)}A(z^{-1}) \\ Q(z) &= A(z) - z^{-(p+1)}A(z^{-1}) \end{aligned}$$

2. Find the roots of $P(z)$ and $Q(z)$, ω_i which are on the unit circle.
3. Exclude trivial roots at $\omega_i = 0$ and $\omega_i = \pi$.

MELP coding

For a 2400 bps:

- 34 bits allocated to scalar quantization of the LSFs
- 8 bits for gain
- 7 bits for pitch and overall voicing
- 5 bits for multi-band voicing
- 1 bit for the jittery state

which is 54 bits. With a frame rate of 22.5 ms, we get an 2400 bps coder.

Outline

Acknowledgments

Most, if not all, figures in this lecture are coming from the book:

T. F. Quatieri: Discrete-Time Speech Signal Processing, principles and practice
2002, Prentice Hall

and have been used after permission from Prentice Hall