ECE 8873
Data Compression & Modeling

Lecture 7: Vector Quantization

School of Electrical and Computer Engineering
Georgia Institute of Technology
Spring, 2004

Notations & Terminology

Quantizer: $Q = (\alpha, \beta, l)$

l is the "length function" which also determines the rate.

Distortion: $d(\mathbf{x}, \hat{\mathbf{x}}) = d(\mathbf{x}, \beta(\alpha(\mathbf{x})))$

Average or expected distortion:

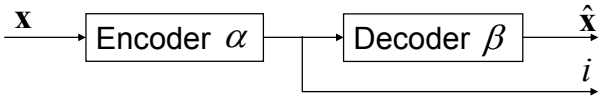
$E[d(\mathbf{X}, \hat{\mathbf{X}})] = E[d(\mathbf{X}, \beta(\alpha(\mathbf{X})))] = D(\alpha, \beta)$

Average or expected rate: $E[l(\alpha(\mathbf{x}))]$

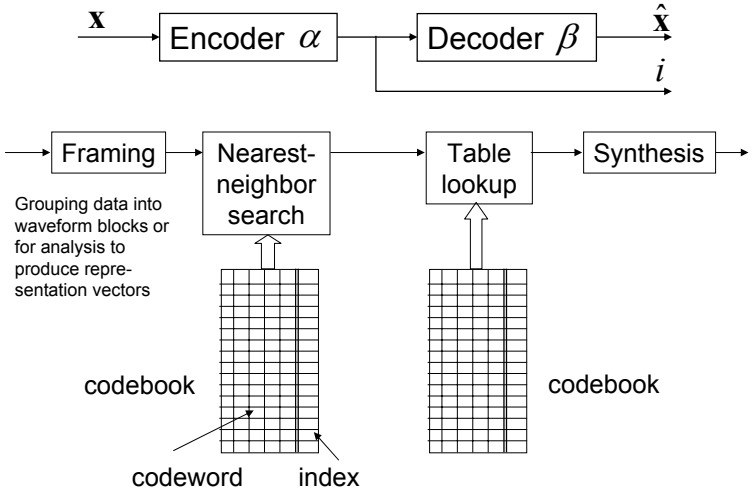
Encoder partition: $S = \{S_i, i \in I\}$

where $S_i = \{\mathbf{x}: \alpha'(\mathbf{x}) = i \in I\}$ $\quad \cup_{i,i\in I} S_i = A$

S_i are disjoint $\Rightarrow S_i \cap S_j = \phi$ if $i \neq j$

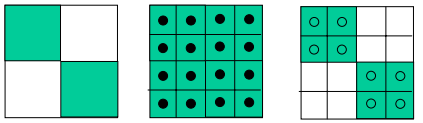


Vector Quantization

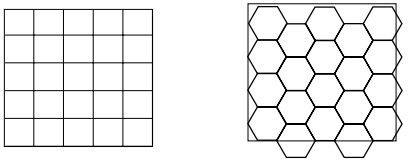


From Scalar to Vector - Advantages

- Gain from statistical dependence



- Gain from optimal space filling polytope



Non-uniform Quantizer

- Coding principle
  - #1: The best representation value in an interval is one that minimizes average distortion in that interval, i.e, the centroid
  - #2: The best interval an input value is assigned to is the one whose centroid is closest to the input value.
- #1 relates to the decoder function, given the encoder results.
- #2 relates to the encoder function, given the decoder design (the codebook).

Empirical Design of Quantizer Codebook

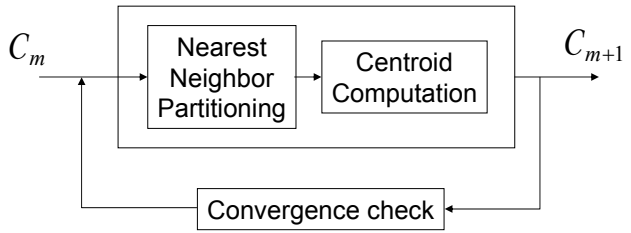
- Use data rather than distribution to design the quantizer – i.e,, train the quantizer.
- Let $\Omega = \{\mathbf{x}_j, j = 1, 2, \cdots, L\}$ be a set of data from a stationary source and
$$Q = (\alpha, \beta), \quad \alpha : \mathbf{x} \to I, \quad \beta : I \to \hat{\mathbf{x}}$$
- The encoder α assigns every data point x_j to a region, indexed i, which contains the subset $\Omega_i = \{\mathbf{x} : \mathbf{x} \in \Omega \text{ and } \alpha(\mathbf{x}) = i \in I\}$ which can be considered a sample of the set $S_i = \{\mathbf{x} : \alpha(\mathbf{x}) = i \in I\}$
- The decoder uses $\{\beta(i), i \in I\}$ as the reproduction vector for each region, incurring $\{\sum_{x \in \Omega_i} d(\mathbf{x}, \beta(i))\}_i$ as the distortion or as the average distortion:
$$(1/L)\sum_{i \in I} \sum_{\mathbf{x} \in \Omega_i} d(\mathbf{x}, \beta(i))$$

Iterative (Hill Climbing) Procedure

$$Q = (\alpha, \beta), \quad \alpha : \mathbf{x} \to I, \quad \beta : I \to \hat{\mathbf{x}}$$

Given the set of reproduction codewords,

$$D(\alpha, \beta) \geq D(\alpha', \beta) \geq D(\alpha', \beta')$$



The Lloyd Algorithm

- Step 1: Get an initial codebook C_m .
- Step 2: Given the codebook C_m , find for each training vector $\mathbf{x} \in \Omega = \{\mathbf{x}_j, j = 1, 2, \cdots, L\}$ the closest codeword and its index $i = q(j)$;
$$i = q(j) = \text{index of the closest codeword for } \mathbf{x}_j$$
- Step 3: After the entire training set is exhausted, compute new centroid for each cell from the set $\Omega_i = \{\mathbf{x}_j; q(j) = i\}, \ i \in I$ respectively;
- Step 4: Compute new average distortion. If it has dropped by only an insignificant amount since last iteration, stop; otherwise, repeat steps 2-4.

Initial Codebook

- Possibilities
  - Random
  - Binary split algorithm
    1. Start with global centroid, set n=1;
    2. Create 2n new codewords from the n (converged) codewords by either perturbing them (multiply each by say .99 and 1.01) or find codewords in each cell that are furthest from each other.
  - M-ary split
  - Other heuristics

Centroid Computation

- Given the distribution $f_X(x)$ and the prescribed distortion measure $d(x,y)$, the centroid is

$$\bar{x} = \arg\min_y \int_X d(x,y) f_X(x) dx$$

With Euclidean distance and empirical data:

$$d(x,\hat{x}) = d_2^2 = (x-\hat{x})^t (x-\hat{x}) \qquad \bar{x} = \frac{1}{L}\sum_{i=1}^{L} x_i$$

With L1 distance and empirical data:

$$d(x,\hat{x}) = |x-\hat{x}| \quad \Rightarrow \quad \text{median}$$

What about other distances or distortion measures?

Lloyd Principle and Consequences

- Suppose all reconstruction values satisfy (Lloyd's) centroid conditions – they minimize **squared error** in respective intervals.
- Then, $\varepsilon = \alpha(X) - X$

$$E[\varepsilon] = 0$$
$$E[\alpha(X)\varepsilon] = 0$$
$$E[\varepsilon^2] = \sigma_\varepsilon^2 = \sigma_X^2 - \sigma_{q(X)}^2$$
$$E[X\varepsilon] = E[(\alpha(X)-\varepsilon)\varepsilon] = -E[\varepsilon^2] = \sigma_\varepsilon^2$$
$$E[\varepsilon^2] = \sigma_\varepsilon^2 = \sigma_X^2 - \sigma_{q(X)}^2 \geq 0 \quad \therefore \sigma_X^2 \geq \sigma_{q(X)}^2$$

Likelihood Distortion

- Itakura-Saito Distortion

$$V(\omega) = \log X(\omega) - \log Y(\omega)$$
$$d_{IS}(\mathbf{x},\mathbf{y}) = \int_{-\pi}^{\pi} \left[e^{V(\omega)} - V(\omega) - 1 \right] \frac{d\omega}{2\pi} = \int_{-\pi}^{\pi} \frac{X(\omega)}{Y(\omega)} \frac{d\omega}{2\pi} - \log\frac{\sigma_{x,\infty}^2}{\sigma_{y,\infty}^2} - 1$$

$\sigma_{x,\infty}^2, \sigma_{y,\infty}^2$ are "1- step" prediction errors of X and Y, respectively

- Linear prediction:

$X(\omega)$ is speech power spectrum and $Y(\omega) = \sigma^2 / |A(e^{j\omega})|^2$

$$d_{IS}\left(X, \frac{\sigma^2}{|A|^2}\right) = \frac{1}{\sigma^2}\int_{-\pi}^{\pi} X(\omega) |A(e^{j\omega})|^2 \frac{d\omega}{2\pi} - \log\frac{\sigma_{x,\infty}^2}{\sigma^2} - 1$$

$$\frac{1}{\sigma^2}\int_{-\pi}^{\pi} X(\omega) |A(e^{j\omega})|^2 \frac{d\omega}{2\pi} = \frac{\mathbf{a}^t \mathbf{R}_p \mathbf{a}}{\sigma^2} \qquad \mathbf{a}^t\mathbf{R}_p\mathbf{a} = r_x(0)r_a(0) + 2\sum_{n=1}^{p} r_x(n) r_a(n)$$

$$r_a(n) = \sum_{i=0}^{p-n} a_i a_{i+n} \quad \text{for } n = 0,1,2,\cdots,p$$

Centroid with Itakura-Saito Distortion

$$d_{IS}\left(X, \frac{\sigma^2}{|A|^2}\right) = \frac{1}{\sigma^2}\int_{-\pi}^{\pi} X(\omega)\,|A(e^{j\omega})|^2\,\frac{d\omega}{2\pi} - \log\frac{\sigma_{x,\infty}^2}{\sigma^2} - 1$$

$$\frac{1}{\sigma^2}\int_{-\pi}^{\pi} X(\omega)\,|A(e^{j\omega})|^2\,\frac{d\omega}{2\pi} = \frac{\mathbf{a}^t\mathbf{R}_p\mathbf{a}}{\sigma^2}$$

$$\Omega_i = \{X:\ X \in \Omega \text{ and } \alpha(X) = i \in I\}$$

$$\hat{D}_i = \sum_{X\in\Omega_i} d_{IS}\left(X, \frac{\sigma^2}{|A|^2}\right) = \frac{1}{\sigma^2}\int_{-\pi}^{\pi}\left(\sum_{X\in\Omega_i} X(\omega)\right)|A(e^{j\omega})|^2\,\frac{d\omega}{2\pi} - \sum_{X\in\Omega_i}\log\frac{\sigma_{x,\infty}^2}{\sigma^2} - \|\Omega_i\|$$

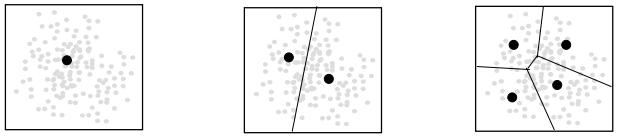
$$\frac{1}{\sigma^2}\int_{-\pi}^{\pi}\left(\sum_{X\in\Omega_i} X(\omega)\right)|A(e^{j\omega})|^2\,\frac{d\omega}{2\pi} = \frac{\sum_{X\in\Omega_i}\mathbf{a}^t\mathbf{R}_{X,p}\mathbf{a}}{\sigma^2} = \frac{\mathbf{a}^t\left(\sum_{X\in\Omega_i}\mathbf{R}_{X,p}\right)\mathbf{a}}{\sigma^2}$$

Same LPC equation but with (average) autocorrelations.

Tree-Structured Vector Quantizers

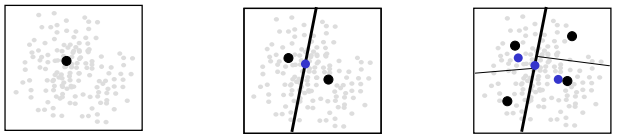
Full VQ with Lloyd algorithm

Each training token is compared to all the codewords in each iteration.

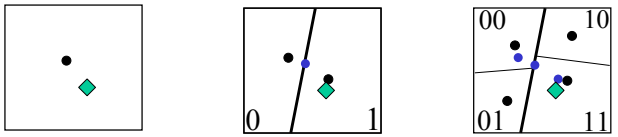


Binary-tree VQ

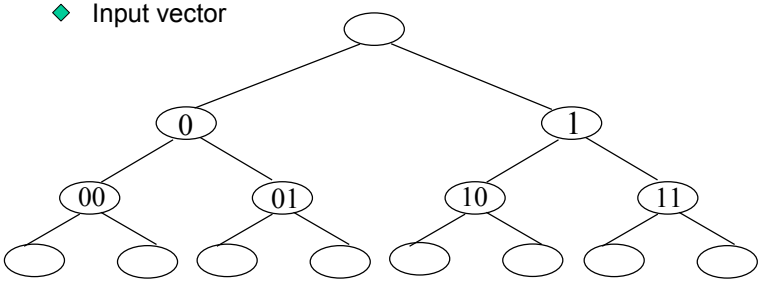
Each training token is compared to two codewords in the respective region during each iteration.



Binary-Tree VQ



◆ Input vector

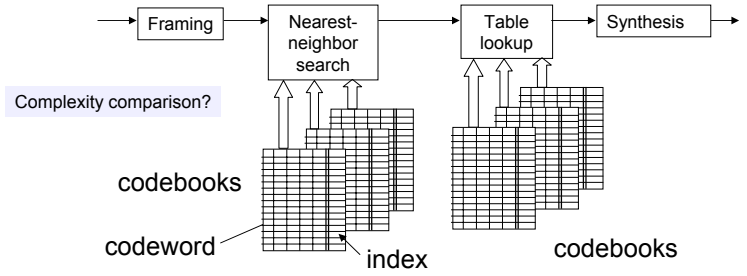


Remarks on Tree-Structured VQ

- Overall distortion performance is suboptimal
- Much faster training due to division of training set (fewer tokens and fewer codewords to compare)
- Larger codebook space in training (almost twice)
- When coupled with code representation and the receiver keeps a copy of the entire tree-structured codebook, it may offer some additional (minor) benefit in robust transmission and recovery of signal (rather than retrieving the codeword of finest resolution, get the intermediate one the received code can reliably allow in the case of bit error or loss).

Vector Quantizer with Product Code

- While going with longer vectors will get closer to the performance bound, the complexity and/or (practical) latency become a problem.
- May need to compromise by using smaller vectors (not to group too many together) or breaking the vectors (even in the case of a complete set of parameters that define a model) into smaller chunks.



Mean-Removed Vector Quantizers

- Often the mean of a vector may have different significance in either statistics or in perception.
- Recall the structural component of information; mean can be considered a 1st order structure.

$$\mathbf{x} = (x_1, x_2, \cdots, x_k) \qquad \mu = \frac{1}{k}\sum_{i=1}^{k} x_i$$

$$\mathbf{x} = (x_1, x_2, \cdots, x_k) = \mu\mathbf{1} + \mathbf{r}$$

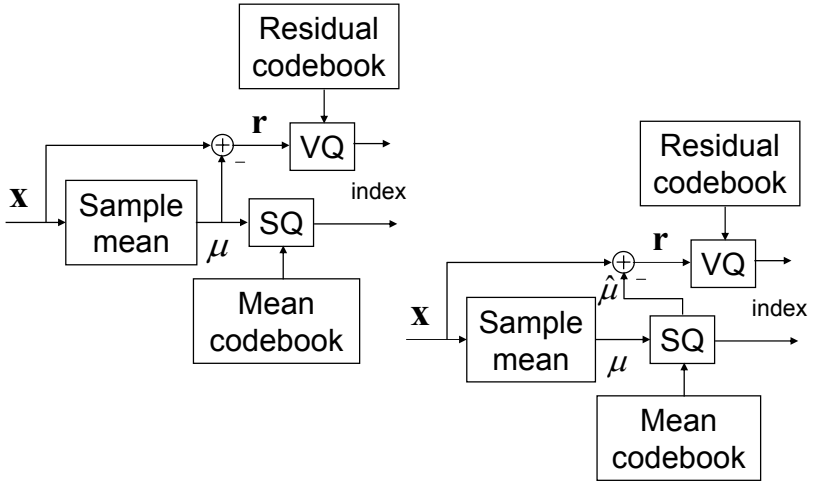
$\mathbf{x} \Rightarrow (\mu, \mathbf{r})$ Code each component "separately."

But $\mathbf{r} = (r_1, r_2, \cdots, r_k)$ with $\mu_r = \frac{1}{k}\sum_{i=1}^{k} r_i = 0$

Therefore, can code the vector $\mathbf{r}$ in k-1 dimensions.

Also consider the statistics of $\Omega_r = \{\mathbf{r}_j, j = 1, 2, \cdots, L\}$

Alternative Structure in M-R VQ



Gain-Shape Vector Quantizers

- Similar to mean, the root mean square value of the vector can be considered a 1st order structure, used to scale the vector.

$$\mathbf{x} = (x_1, x_2, \cdots, x_k) \qquad g(\mathbf{x}) = \sqrt{\sum_{i=1}^{k} x_i^2}$$

$\mathbf{s} = \mathbf{x}\, g^{-1}(\mathbf{x})$ is the shape vector

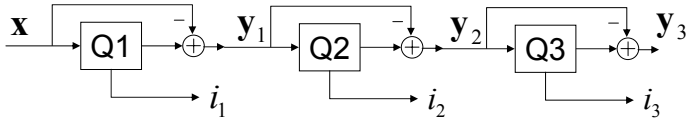
$\mathbf{x} \Rightarrow (g, \mathbf{s})$ Code each component separately.

Distortion is

$$d(\mathbf{x}, \hat{g}\,\hat{\mathbf{s}}) = \| \mathbf{x} - \hat{g}\,\hat{\mathbf{s}} \|^2 = \| \mathbf{x} \|^2 + \hat{g}^2 - 2\hat{g}(\mathbf{x}^t\hat{\mathbf{s}})$$

- Select $\hat{\mathbf{s}}$ to maximize $\mathbf{x}^t\hat{\mathbf{s}}$
- Select $\hat{g}$ to minimize $(\hat{g} - \mathbf{x}^t\hat{\mathbf{s}})^2$

Multi-Stage Vector Quantizers



$$\mathbf{y}_1 = \mathbf{x} - Q_1(\mathbf{x})$$
$$\mathbf{y}_2 = \mathbf{y}_1 - Q_2(\mathbf{y}_1)$$
$$\mathbf{y}_3 = \mathbf{y}_2 - Q_3(\mathbf{y}_2)$$

$$Q(\mathbf{x}) = Q_1(\mathbf{x}) + Q_2(\mathbf{y}_1) + Q_3(\mathbf{y}_2) + \cdots$$
$$\mathbf{x} - Q(\mathbf{x}) = \mathbf{x} - [Q_1(\mathbf{x}) + Q_2(\mathbf{y}_1) + Q_3(\mathbf{y}_2) + \cdots]$$
$$= \mathbf{y}_1 - [Q_2(\mathbf{y}_1) + Q_3(\mathbf{y}_2) + \cdots] = \cdots$$
$$= \mathbf{y}_{last-1} - Q_{last}(\mathbf{y}_{last-1})$$

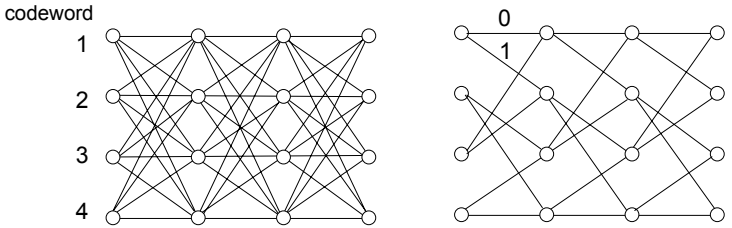
- Since the quantization error variance never grows, the error shrinks at higher stages.
- Potential in progressive or hierarchical schemes for layered coding

Finite State Vector Quantizer

- A quantization/coding scheme consists of
  - An encoder $\alpha: \ A^* \to B^* = \{0,1\}^*$
  - A decoder $\beta: \ B^* \to \hat{A}^*, \ \beta(\alpha(x)) = \hat{x} \in \hat{A}$
  - A reproduction codebook $\hat{A} = \{\beta(i), \ i \in I\}$
- A finite state quantization/coding scheme consists of:
  - An encoder $\alpha: \ (A^*, S^*) \to B^* = \{0,1\}^*$
    - With explicit sequential dependence: $b_t = \alpha(x_t, s_t)$
  - A decoder $\hat{x}_t = \beta(s_t, b_t)$
  - A reproduction codebook $\hat{A} = \{\beta(j,i), \ j \in I_s, i \in I_b\}$
  - A next state function: $s_{t+1} = g(s_t, b_t)$

Trellis or Finite State Vector Quantizer

- Explore sequential dependency between symbols.
- Nearest neighbor search under trellis constraint; rest of the Lloyd algorithm applies.



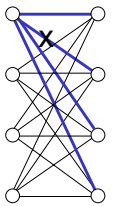
2-bit codebook
Full rate transmission

2-bit codebook
1-bit transmission

Trellis can be implemented with a shift register. ....0110 →

Trellis Design by Pruning

- How to construct the trellis? Many heuristics.
- Consider pruning from the full-rate case – minimum degradation pruning.



Training **sequence**: $\Omega = \{\mathbf{x}_j, j = 1, 2, \cdots, L\}$

For each $\mathbf{x} \in \Omega_i = \{\mathbf{x}_j; q(j) = i\}, \ i \in I$

define $\Omega_{ik} = \{\mathbf{x}_{j+1}; \mathbf{x}_j \in \Omega_i, q(j+1) = k\}, \ k \in I'$

$$\varsigma(k) = \sum_{\mathbf{x} \in \Omega_{ik}} [d(\mathbf{x}, \beta(k')) - d(\mathbf{x}, \beta(k))]$$

$q(j+1) = k$ is the index of the closest codeword

$q'(j+1) = k'$ is the index of second closest codeword

$\varsigma(k)$ represents the degradation if the connection between i and k is broken.

Find $\underline{k} = \arg\min_k \varsigma(k)$; break $i \to \underline{k}$; repeat Lloyd iteration to update codewords; continue to prune till desired rate.

Performance - PTVQ