Vector Quantization
Performance of Scalar Quantizers: Distortion (MSE) and Bit Rate

\[ D = \mathbb{E}\left\{ (S - Q(S))^2 \right\} = \sum_{\forall k} \int_{u_k}^{u_{k+1}} (s - s'_k)^2 f(s) \, ds \]

\[ R = \mathbb{E}\left\{ \ell(Q(S)) \right\} = \sum_{\forall k} \ell_k \int_{u_k}^{u_{k+1}} f(s) \, ds \]
Lloyd Quantizer

- Minimizes distortion $D$ for given number $K$ of reconstruction levels
- Two optimization criterions:
  - Centroid condition (MSE): $s_k' = \mathbb{E}\{ S \mid S \in I_k \}$
  - Nearest neighbor condition (MSE): $u_k = (s_k' + s_{k-1}')/2$
- Lloyd quantizer design: Iterate between the two optimization criterions
Lloyd Quantizer

- Minimizes distortion $D$ for given number $K$ of reconstruction levels
- Two optimization criterions:
  - Centroid condition (MSE): $s'_k = \mathbb{E}\{ S \mid S \in \mathcal{I}_k \}$
  - Nearest neighbor condition (MSE): $u_k = (s'_k + s'_{k-1})/2$
- Lloyd quantizer design: Iterate between the two optimization criterions

Entropy-Constrained Lloyd Quantizer

- Minimizes rate-distortion cost $J = D + \lambda R$ for given Lagrange multiplier $\lambda > 0$
- Three optimization criterions:
  - Centroid condition (MSE): $s'_k = \mathbb{E}\{ S \mid S \in \mathcal{I}_k \}$
  - Entropy condition:
    $$\ell_k = -\log_2 \int_{u_k}^{u_k+1} f(s) \, ds$$
  - Mod. nearest neighbor condition (MSE):
    $$u_k = (s'_k + s'_{k-1})/2 + (\lambda/2)(\ell_k - \ell_{k-1})/(s'_k + s'_{k-1})$$
- EC-Lloyd quantizer design: Iterate between the optimization criterions
Last Lectures: Performance of Scalar Quantizers

High-Rate Approximations (MSE Distortion)

- General form of high-rate distortion-rate function

\[ D_X(R) = \varepsilon_X^2 \cdot \sigma^2 \cdot 2^{-2R} \]

where the constant factor \( \varepsilon_X^2 \) depends on shape of pdf and quantizer design

- Lloyd + fixed length:
  \[ \varepsilon_F^2 = \frac{1}{12} \left( \int_{-\infty}^{\infty} 3 \sqrt{f(s/\sigma)} \, ds \right)^3 \]

- EC-Lloyd + variable length:
  \[ \varepsilon_V^2 = \frac{1}{12} 2^2 h(S/\sigma) \quad \text{with} \quad h(S) = -\int_{-\infty}^{\infty} f(s) \log_2 f(s) \, ds \]

- Shannon lower bound:
  \[ \varepsilon_L^2 = \frac{1}{2\pi e} 2^2 h(S/\sigma) \]

Comparison of Coding Efficiency

EC-Lloyd often significantly better than Lloyd (Gauss: 2.81 dB; Laplace: 5.63 dB)

Constant high-rate performance gap between EC-Lloyd and Shannon lower bound

\[ D_V(R) - D_L(R) = \frac{1}{2} \log_2 \pi e \approx 0.25 \]
**Last Lectures: Performance of Scalar Quantizers**

**High-Rate Approximations (MSE Distortion)**

- General form of high-rate distortion-rate function

\[
D_X(R) = \varepsilon_X^2 \cdot \sigma^2 \cdot 2^{-2R}
\]

where the constant factor \(\varepsilon_X^2\) depends on shape of pdf and quantizer design

- Lloyd + fixed length:
  \[
  \varepsilon_F^2 = \frac{1}{12} \left( \int_{-\infty}^{\infty} 3 \sqrt{f(s/\sigma)} \, ds \right)^3
  \]

- EC-Lloyd + variable length:
  \[
  \varepsilon_V^2 = \frac{1}{12} 2^2 h(S/\sigma) \quad \text{with} \quad h(S) = -\int_{-\infty}^{\infty} f(s) \log_2 f(s) \, ds
  \]

- Shannon lower bound:
  \[
  \varepsilon_L^2 = \frac{1}{2\pi e} 2^2 h(S/\sigma)
  \]

**Comparison of Coding Efficiency**

- EC-Lloyd often significantly better than Lloyd (Gauss: 2.81 dB; Laplace: 5.63 dB)

- Constant high-rate performance gap between EC-Lloyd and Shannon lower bound

\[
\frac{D_V(R)}{D_L(R)} = \frac{\pi e}{6} \approx 1.42 \quad (1.53 \, \text{dB}), \quad R_V(D) - R_L(D) = \frac{1}{2} \log_2 \frac{\pi e}{6} \approx 0.25
\]
Uniform Reconstruction Quantizers (URQs)

- Simple decoding process: \( s'_n = \Delta \cdot q_n \)
- Encoder can choose trade-off between coding efficiency and complexity
  - Simplest encoding: \( q_n = \text{round}\left(\frac{s_n}{\Delta}\right) \)
  - Optimal encoding: Choose \( q_n \) that minimizes Lagrange cost \( J(q_n) = (s_n - q_n \cdot \Delta)^2 + \lambda \cdot \ell_k \), typically using fixed relationship \( \lambda = \text{const} \cdot \Delta^2 \)

- URQs with optimal encoding are virtually as good as optimal scalar quantizers (for typical pdfs)
Quantization: Open Questions

Performance Gap to Theoretical Bound

- Remember: High-rate performance of optimal scalar quantizer for IID sources

\[ \frac{D_V}{D_L}(R) = \frac{\pi e}{6} \approx 1.42 \quad (1.53 \text{ dB loss in SNR}) \]

- What causes this performance gap?
- How can the quantizer performance be improved?
Quantization: Open Questions

Performance Gap to Theoretical Bound

- Remember: High-rate performance of optimal scalar quantizer for IID sources

\[
\frac{D_V}{D_L}(R) = \frac{\pi e}{6} \approx 1.42 \quad (1.53 \text{ dB loss in SNR})
\]

→ What causes this performance gap?
→ How can the quantizer performance be improved?

Quantization of Sources with Memory

- Scalar quantizers cannot exploit dependencies between samples (use only marginal pdf)
→ How can we improve lossy coding for sources with memory?
Quantization: Open Questions

Performance Gap to Theoretical Bound
- Remember: High-rate performance of optimal scalar quantizer for IID sources

\[
\frac{D_V}{D_L}(R) = \frac{\pi e}{6} \approx 1.42 \quad (1.53 \text{ dB loss in SNR})
\]

- What causes this performance gap?
- How can the quantizer performance be improved?

Quantization of Sources with Memory
- Scalar quantizers cannot exploit dependencies between samples (use only marginal pdf)
- How can we improve lossy coding for sources with memory?
  - Conditional entropy coding of quantization indexes?
  - Combination of scalar quantization and prediction?
  - ...?
Scalar Quantizers in $N$-dimensional Signal Space

Interpretation of Scalar Quantization in $N$-dimensional Signal Space

$N$-dimensional input vector $s$ is mapped to $N$-dimensional reconstructed vector $s'$. All vectors $s$ inside a quantization cell $C_k$ are mapped to the same reconstruction vector $s'_k$. Quantization cells $C_k$ form hyper-rectangles in $N$-dimensional signal space. Reconstruction vectors $s'_k$ lie on orthogonal grid aligned with coordinate axes.
Scalar Quantizers in $N$-dimensional Signal Space

Interpretation of Scalar Quantization in $N$-dimensional Signal Space

- $N$-dimensional input vector $s$ is mapped to $N$-dimensional reconstructed vector $s'$
- All vectors $s$ inside a quantization cell $C_k$ are mapped to the same reconstruction vector $s'_k$
Scalar Quantizers in \( N \)-dimensional Signal Space

**Interpretation of Scalar Quantization in \( N \)-dimensional Signal Space**

- \( N \)-dimensional input vector \( s \) is mapped to \( N \)-dimensional reconstructed vector \( s' \)
- All vectors \( s \) inside a quantization cell \( C_k \) are mapped to the same reconstruction vector \( s'_k \)
- \( \rightarrow \) Quantization cells \( C_k \) form hyper-rectangles in \( N \)-dimensional signal space
Scalar Quantizers in $N$-dimensional Signal Space

**Interpretation of Scalar Quantization in $N$-dimensional Signal Space**

- $N$-dimensional input vector $s$ is mapped to $N$-dimensional reconstructed vector $s'$
- All vectors $s$ inside a quantization cell $C_k$ are mapped to the same reconstruction vector $s'_k$

- Quantization cells $C_k$ form hyper-rectangles in $N$-dimensional signal space
- Reconstruction vectors $s'_k$ lie on orthogonal grid aligned with coordinate axes
Vector Quantization: Relaxing Structural Constraints

**Scalar quantizer**

(dimension $N = 1$)
Vector Quantization: Relaxing Structural Constraints

**scalar quantizer**
(dimension $N = 1$)

---

**Vector Quantization**

- Joint quantization of vectors/blocks $s$ of $N > 1$ successive input samples
Vector Quantization: Relaxing Structural Constraints

scalar quantizer
(dimension $N = 1$)

vector quantizer
(dimension $N = 2$)

**Vector Quantization**
- Joint quantization of vectors(blocks $s$) of $N > 1$ successive input samples
Vector Quantization: Relaxing Structural Constraints

**Scalar Quantizer**

(dimension $N = 1$)

**Vector Quantizer**

(dimension $N = 2$)

**Vector Quantization**

- Joint quantization of vectors/blocks $s$ of $N > 1$ successive input samples
- Relax structural constraints that are implicitly imposed by scalar quantization
  - Quantization cells $C_k$ can be arbitrarily shaped in $N$-dimensional space
  - Reconstruction vectors $s'_k$ can be arbitrarily placed in $N$-dimensional space
Vector Quantization: Relaxing Structural Constraints

**scalar quantizer**
(dimension $N = 1$)

**vector quantizer**
(dimension $N = 2$)

Vector Quantization

- Joint quantization of vectors/blocks $s$ of $N > 1$ successive input samples
- Relax structural constraints that are implicitly imposed by scalar quantization
  - Quantization cells $C_k$ can be arbitrarily shaped in $N$-dimensional space
  - Reconstruction vectors $s_k'$ can be arbitrarily placed in $N$-dimensional space

→ Allows a number of new options in designing quantizers
**Structure of Vector Quantizers**

**Vector Quantizers of Quantizer Dimension $N$**

- Map $N$-d input vectors $s$ to $N$-d output vectors $s'_k$

\[
Q : \mathbb{R}^N \mapsto \{ s'_0, s'_1, s'_2, \ldots \}
\]

- Partition $N$-d space into countable number of quantization cells $C_k$

\[
C_k = \{ s \in \mathbb{R}^N : Q(s) = s'_k \}
\]

- All input vectors $s$ that fall inside a quantization cell $C_k$ are mapped to the associated reconstruction vector $s'_k$
Structure of Vector Quantizers

Vector Quantizers of Quantizer Dimension $N$
- Map $N$-d input vectors $s$ to $N$-d output vectors $s'_k$
  $$Q : \mathbb{R}^N \mapsto \{ s'_0, s'_1, s'_2, \ldots \}$$
- Partition $N$-d space into countable number of quantization cells $C_k$
  $$C_k = \{ s \in \mathbb{R}^N : Q(s) = s'_k \}$$
- All input vectors $s$ that fall inside a quantization cell $C_k$ are mapped to the associated reconstruction vector $s'_k$

Vector Quantization and Entropy Coding
- Quantization index $k$ indicates quantization cell $C_k$ and reconstruction vector $s'_k$
  - Encoder mapping: $\alpha(s) = k, \ \forall s \in C_k$
  - Decoder mapping: $\beta(k) = s'_k$
Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $C_k$ are difficult to store and check
- Concept of quantization cells is not required in practice

Possible distance measures:

- **MSE distortion:**
  \[ d = \| s - s_k' \|^2 \]

- **Lagrangian cost:**
  \[ d = \| s - s_k' \|^2 + \lambda \ell_k \]
Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $C_k$ are difficult to store and check
- Concept of quantization cells is not required in practice

Encoding
- Select the reconstruction vector $s'_k$ that minimizes a distance measure $d$ to the input vector $s$
Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $C_k$ are difficult to store and check
- Concept of quantization cells is not required in practice

**Encoding**

- Select the reconstruction vector $s'_k$ that minimizes a **distance measure** $d$ to the input vector $s$
- Possible distance measures:
  - MSE distortion: $d = ||s - s'_k||_2^2$
  - Lagrangian cost: $d = ||s - s'_k||_2^2 + \lambda \ell_k$
Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $C_k$ are difficult to store and check
- Concept of quantization cells is not required in practice

Encoding

- Select the reconstruction vector $s'_k$ that minimizes a distance measure $d$ to the input vector $s$
- Possible distance measures:
  - MSE distortion: $d = ||s - s'_k||^2_2$
  - Lagrange cost: $d = ||s - s'_k||^2_2 + \lambda \ell_k$

Decoding

- Output reconstruction vector $s'_k$ indicated by transmitted quantization index $k$ (use array in decoder)
Performance of Vector Quantizers: Bit Rate

- Let $\ell_k$ be the codeword length for quantization index $k$

**Average bit rate $R$ per sample**

$$ R = \frac{1}{N} \mathbb{E}\{ \ell( Q(S) ) \} = \frac{1}{N} \sum_{\forall k} p_k \ell_k $$
Performance of Vector Quantizers: Bit Rate

- Let $\ell_k$ be the codeword length for quantization index $k$

→ **Average bit rate $R$ per sample**

\[
R = \frac{1}{N} \mathbb{E}\left\{ \ell(Q(S)) \right\} = \frac{1}{N} \sum_{\forall k} p_k \ell_k
\]

- Probability $p_k$ of quantization cell $C_k$ / quant. index $k$

\[
p_k = \int_{C_k} f(s) \, ds
\]
Performance of Vector Quantizers: Bit Rate

- Let $\ell_k$ be the codeword length for quantization index $k$

$\Rightarrow$ **Average bit rate $R$ per sample**

$$R = \frac{1}{N} \mathbb{E}\left\{ \ell(Q(S)) \right\} = \frac{1}{N} \sum_{\forall k} p_k \ell_k$$

- Probability $p_k$ of quantization cell $C_k$ / quant. index $k$

$$p_k = \int_{C_k} f(s) \, ds$$

- Approximation for training set

$$p_k = \frac{n(k)}{\sum_k n(k)}$$

where $n(k)$ is the number of vectors assigned to $C_k$ / $s'_k$
Performance of Vector Quantizers: Distortion

Average MSE distortion $D$ per sample

$$D = \frac{1}{N} E \left\{ \| S - Q(S) \|_2^2 \right\}$$
Performance of Vector Quantizers: Distortion

Average MSE distortion $D$ per sample

$$D = \frac{1}{N} E\left\{ \| S - Q(S) \|_2^2 \right\}$$

$$= \frac{1}{N} \int_{\mathbb{R}^N} \| s - Q(s) \|_2^2 f(s) \, ds$$
Performance of Vector Quantizers: Distortion

→ Average MSE distortion $D$ per sample

$$D = \frac{1}{N} \mathbb{E}\left\{ \| S - Q(S) \|_2^2 \right\}$$

$$= \frac{1}{N} \int_{\mathbb{R}^N} \| s - Q(s) \|_2^2 f(s) \, ds$$

$$= \frac{1}{N} \sum_{\forall k} \int_{C_k} \| s - s'_k \|_2^2 f(s) \, ds$$
Performance of Vector Quantizers: Distortion

Average MSE distortion $D$ per sample

$$D = \frac{1}{N} E \left\{ \| S - Q(S) \|^2 \right\}$$

$$= \frac{1}{N} \int_{\mathbb{R}^N} \| s - Q(s) \|^2 f(s) \, ds$$

$$= \frac{1}{N} \sum_{\forall k} \int_{C_k} \| s - s'_k \|^2 f(s) \, ds$$

$$D = \frac{1}{N} \sum_{\forall k} \int_{C_k} (s - s'_k)^T (s - s'_k) f(s) \, ds$$
Performance of Vector Quantizers: Distortion

→ Average MSE distortion $D$ per sample

\[
D = \frac{1}{N} \mathbb{E} \left\{ \| S - Q(S) \|^2 \right\}
\]

\[
= \frac{1}{N} \int_{\mathbb{R}^N} \| s - Q(s) \|^2 f(s) \, ds
\]

\[
= \frac{1}{N} \sum_{\forall k} \int_{C_k} \| s - s'_k \|^2 f(s) \, ds
\]

\[
D = \frac{1}{N} \sum_{\forall k} \int_{C_k} (s - s'_k)^T (s - s'_k) f(s) \, ds
\]

- Approximation for training set $\{s_n\}$ of $L$ vectors

\[
D = \frac{1}{L} \sum_{\forall n} \| s_n - Q(s_n) \|^2
\]
Optimal Vector Quantizer for Fixed-Length Coding

Goal: Minimize MSE Distortion for $K$ Quantization Cells

- Similar to Scalar Lloyd Quantizer
- Neglect impact of entropy coding \(\rightarrow\) Consider fixed-length coding

\[
R = \frac{1}{N} \left\lceil \log_2 K \right\rceil \quad \text{(typically } K = 2^B, \text{ with } B \text{ being the bits per codeword)}
\]

\[
D = \frac{1}{N} \sum_{\forall k} \int_{C_k} \left\| s - s'_{k} \right\|^2 f(s) \, ds
= \frac{1}{N} \sum_{\forall k} \int_{C_k} (s - s'_{k})^T (s - s'_{k}) f(s) \, ds
\]
Optimal Vector Quantizers with Fixed-Length Coding

Goal: Minimize MSE Distortion for $K$ Quantization Cells

- Similar to Scalar Lloyd Quantizer
- Neglect impact of entropy coding  $\Rightarrow$ Consider fixed-length coding
- Rate $R$ and MSE distortion $D$ are given by

$$R = \frac{1}{N} \left\lceil \log_2 K \right\rceil$$ (typically $K = 2^B$, with $B$ being the bits per codeword)

$$D = \frac{1}{N} \sum_{\forall k} \int_{C_k} \| s - s'_k \|_2^2 f(s) \, ds = \frac{1}{N} \sum_{\forall k} \int_{C_k} (s - s'_k)^T (s - s'_k) f(s) \, ds$$
Optimal Vector Quantizers with Fixed-Length Coding

Optimal Vector Quantizer for Fixed-Length Coding

Goal: Minimize MSE Distortion for $K$ Quantization Cells

- Similar to Scalar Lloyd Quantizer
- Neglect impact of entropy coding $\Rightarrow$ Consider fixed-length coding
- Rate $R$ and MSE distortion $D$ are given by

$$R = \frac{1}{N} \left\lceil \log_2 K \right\rceil$$
(typically $K = 2^B$, with $B$ being the bits per codeword)

$$D = \frac{1}{N} \sum_k \int_{C_k} \left\| s - s_k' \right\|^2_2 f(s) \, ds = \frac{1}{N} \sum_k \int_{C_k} (s - s_k')^T (s - s_k') f(s) \, ds$$

Optimize Quantizer of size $K$

- Derive necessary conditions for optimality (similar to Lloyd quantizer)
- Construct iterative algorithm for designing quantizer
Optimality Conditions for Fixed-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $s'_k$)

$$s'_k = \mathbb{E}\{ S \mid S \in C_k \} = \frac{1}{p_k} \int_{C_k} s f(s) \, ds$$
Optimality Conditions for Fixed-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $s_k'$)

$$s'_k = \mathbb{E}\{ S \mid S \in C_k \} = \frac{1}{p_k} \int_{C_k} s f(s) \, ds$$

Centroid condition for training set

$$s'_k = \frac{1}{n(k)} \sum_{\forall s: \alpha(s)=k} s$$

with

$$n(k) = \sum_{\forall s: \alpha(s)=k} 1$$
Optimality Conditions for Fixed-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1 Centroid condition (for reconstruction vectors $s'_k$)

$$s'_k = \mathbb{E}\{ S \mid S \in C_k \} = \frac{1}{p_k} \int_{C_k} s f(s) \, ds$$

→ Centroid condition for training set

$$s'_k = \frac{1}{n(k)} \sum_{\forall s: \alpha(s)=k} s \quad \text{with} \quad n(k) = \sum_{\forall s: \alpha(s)=k} 1$$

2 Nearest neighbour condition (for quantization cells $C_k$ / encoder mapping $\alpha(.)$)

$$\alpha(s) = \arg \min_{\forall k} \| s - s'_k \|_2^2$$
The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension $N$ and the size $K$ of the quantizer
- a sufficiently large realization $\{s_n\}$ of considered source
The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension $N$ and the size $K$ of the quantizer
- a sufficiently large realization $\{s_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$
The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension $N$ and the size $K$ of the quantizer
- a sufficiently large realization $\{s_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$
2. Associate all vectors of the training set $\{s_n\}$ with one of the quantization cells $C_k$

$$q(s_n) = \arg \min_{\forall k} \|s_n - s'_k\|^2_2$$  (nearest neighbor condition)
The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension $N$ and the size $K$ of the quantizer
- a sufficiently large realization $\{s_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$

2. Associate all vectors of the training set $\{s_n\}$ with one of the quantization cells $C_k$

   $$ q(s_n) = \arg \min_k \| s_n - s'_k \|_2^2 $$  \hspace{1cm} (nearest neighbor condition)

3. Update the reconstruction vectors $\{s'_k\}$ according to

   $$ s'_k = \frac{1}{n(k)} \sum_{\forall n: q(s_n) = k} s_n $$ \hspace{1cm} (centroid condition)

   where $n(k)$ is the number of sample vectors $s_n$ assigned to $C_k$
The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension $N$ and the size $K$ of the quantizer
- a sufficiently large realization $\{s_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$
2. Associate all vectors of the training set $\{s_n\}$ with one of the quantization cells $C_k$

$$q(s_n) = \arg \min_k \|s_n - s'_k\|_2^2$$
(nearest neighbor condition)

3. Update the reconstruction vectors $\{s'_k\}$ according to

$$s'_k = \frac{1}{n(k)} \sum_{\forall n: q(s_n) = k} s_n$$
(centroid condition)

where $n(k)$ is the number of sample vectors $s_n$ assigned to $C_k$

4. Repeat the previous two steps until convergence
Example: LBG Algorithm for Gaussian IID \((\sigma^2 = 1, N = 2, K = 16)\)

\[
\begin{align*}
R &= 2 \\
D &= 0.122 \\
\text{SNR} &= 9.12 \text{dB}
\end{align*}
\]
Example: LBG Algorithm for Gaussian IID \((\sigma^2 = 1, N = 2, K = 16)\)

- **Initialization**
  \[ R = 2 \]
  \[ D = 0.122 \]
  \[ \text{SNR} = 9.12 \text{dB} \]

- **After iteration 5**
  \[ R = 2 \]
  \[ D = 0.117 \]
  \[ \text{SNR} = 9.31 \text{dB} \]
Example: LBG Algorithm for Gaussian IID \((\sigma^2 = 1, N = 2, K = 16)\)

 Initialization

\[
\begin{array}{c}
R = 2 \\
D = 0.122 \\
\text{SNR} = 9.12 \text{dB}
\end{array}
\]

After iteration 5

\[
\begin{array}{c}
R = 2 \\
D = 0.117 \\
\text{SNR} = 9.31 \text{dB}
\end{array}
\]

After iteration 15

\[
\begin{array}{c}
R = 2 \\
D = 0.114 \\
\text{SNR} = 9.43 \text{dB}
\end{array}
\]
Example: LBG Algorithm for Gaussian IID \((\sigma^2 = 1, N = 2, K = 16)\)

- **Initialization**
  - \(R = 2\)
  - \(D = 0.122\)
  - \(\text{SNR} = 9.12 \text{dB}\)

- **After iteration 5**
  - \(R = 2\)
  - \(D = 0.117\)
  - \(\text{SNR} = 9.31 \text{dB}\)

- **After iteration 15**
  - \(R = 2\)
  - \(D = 0.114\)
  - \(\text{SNR} = 9.43 \text{dB}\)

- **Final result**
  - \(R = 2\)
  - \(D = 0.107\)
  - \(\text{SNR} = 9.69 \text{dB}\)
Comparison to Scalar Quantization: Gaussian IID ($\sigma^2 = 1$, $R = 2$)

Lloyd ($N = 1$)

\[ D = 0.117 \]
\[ \text{SNR} = 9.30 \text{dB} \]

LBG ($N = 2$)

\[ D = 0.107 \]
\[ \text{SNR} = 9.69 \text{dB} \]

→ Improvement of $0.39 \text{ dB}$ (distortion reduction by factor 0.91)
Comparison to Scalar Quantization: Gaussian IID \((\sigma^2 = 1, \ R = 4)\)

\[\text{Lloyd (} N = 1 \text{)}\]

- \(D = 0.00951\)
- \(\text{SNR} = 20.22\, \text{dB}\)

\[\text{LBG (} N = 2 \text{)}\]

- \(D = 0.00767\)
- \(\text{SNR} = 21.15\, \text{dB}\)

\(\rightarrow \text{Improvement of 0.93 dB}\) (distortion reduction by factor 0.81)
Comparison to Scalar Quantization: Laplacian IID \((\sigma^2 = 1, R = 2)\)

- **Lloyd** \((N = 1)\):
  - \(D = 0.176\)
  - SNR = 7.54 dB

- **LBG** \((N = 2)\):
  - \(D = 0.129\)
  - SNR = 8.89 dB

→ Improvement of 1.35 dB (distortion reduction by factor 0.73)
Comparison to Scalar Quantization: Laplacian IID \((\sigma^2 = 1, R = 4)\)

**Lloyd \((N = 1)\)**

- \(D = 0.0153\)
- \(\text{SNR} = 18.14\,\text{dB}\)

**LBG \((N = 2)\)**

- \(D = 0.0098\)
- \(\text{SNR} = 20.08\,\text{dB}\)

**Improvement of 1.94 dB** (distortion reduction by factor 0.64)
The Vector Quantizer Advantage

Gain over scalar quantization can be assigned to 3 effects:
Gain over scalar quantization can be assigned to 3 effects:

- **Space filling advantage:**
  - \( \mathbb{Z}^N \) lattice is not most efficient sphere packing in \( N \) dimensions (\( N > 1 \))
  - Independent from source distribution or statistical dependencies
  - Maximum gain for \( N \to \infty \): 1.53 dB
The Vector Quantizer Advantage

Gain over scalar quantization can be assigned to 3 effects:

- **Space filling advantage:**
  - $\mathbb{Z}^N$ lattice is not most efficient sphere packing in $N$ dimensions ($N > 1$)
  - Independent from source distribution or statistical dependencies
  - Maximum gain for $N \rightarrow \infty$: 1.53 dB

- **Shape advantage:**
  - Exploit shape of source pdf
  - Can also be exploited using entropy-constrained scalar quantization
The Vector Quantizer Advantage

Gain over scalar quantization can be assigned to 3 effects:

- **Space filling advantage:**
  - \( \mathbb{Z}^N \) lattice is not most efficient sphere packing in \( N \) dimensions (\( N > 1 \))
  - Independent from source distribution or statistical dependencies
  - Maximum gain for \( N \to \infty \): 1.53 dB

- **Shape advantage:**
  - Exploit shape of source pdf
  - Can also be exploited using entropy-constrained scalar quantization

- **Memory advantage:**
  - Exploit statistical dependencies of the source
  - Can also be exploited using predictive coding, transform coding, block entropy coding or conditional entropy coding
Space-Filling Advantage: LBG for Uniform IID Source

**Lloyd** \( (N = 1) \): SNR = 23.97 dB

**LBG** \( (N = 2) \): SNR = 24.14 dB

- LBG algorithm approaches approximate hexagonal lattice
- Improvement of 0.17 dB
Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- Space filling gain: Densest packing of “optimal” quantization cells in signal space
- MSE distortion: Densest packing of spheres in $N$-dimensional space
Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- Space filling gain: Densest packing of “optimal” quantization cells in signal space
- MSE distortion: Densest packing of spheres in $N$-dimensional space
Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- Space filling gain: Densest packing of “optimal” quantization cells in signal space
- MSE distortion: **Densest packing of spheres in $N$-dimensional space**
Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- Space filling gain: Densest packing of “optimal” quantization cells in signal space
- MSE distortion: Densest packing of spheres in $N$-dimensional space
The Vector Quantizer Advantage / Space-Filling Advantage

Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- **Space filling gain:** Densest packing of “optimal” quantization cells in signal space

- **MSE distortion:** *Densest packing of spheres in $N$-dimensional space*
Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- Space filling gain: Densest packing of "optimal" quantization cells in signal space
- MSE distortion: Densest packing of spheres in $N$-dimensional space
Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- Space filling gain: Densest packing of “optimal” quantization cells in signal space
- MSE distortion: Densest packing of spheres in $N$-dimensional space
  - 2 dimensions: Hexagonal lattice (like honeycombs)
Space-Filling Advantage: Sphere Packing in $N$-dimensional Signal Space

- Space filling gain: Densest packing of “optimal” quantization cells in signal space
- MSE distortion: Densest packing of spheres in $N$-dimensional space
  - 2 dimensions: Hexagonal lattice (like honeycombs)
  - 3 dimensions: Cuboidal lattice (stapling of cannon balls / oranges)
Space-Filling Advantage: Sphere Packing Density

Center density

- Consider $N$-dimensional spheres with radius $r = 1$
Space-Filling Advantage: Sphere Packing Density

Center density
- Consider $N$-dimensional spheres with radius $r = 1$
- Measure for packing density: Center density

$$\delta = \frac{\text{average number of sphere centers}}{\text{unit volume}}$$
Space-Filling Advantage: Sphere Packing Density

Center density

- Consider \( N \)-dimensional spheres with radius \( r = 1 \)
- Measure for packing density: **Center density**

\[
\delta = \frac{\text{average number of sphere centers}}{\text{unit volume}}
\]

- Example: \( N = 1 \) (SQ with intervals of size \( 2r = 2 \))

\[
\delta = \frac{1}{2}
\]
Space-Filling Advantage: Sphere Packing Density

Center density
- Consider \( N \)-dimensional spheres with radius \( r = 1 \)
- Measure for packing density: **Center density**

\[
\delta = \frac{\text{average number of sphere centers}}{\text{unit volume}}
\]

- Example: \( N = 1 \) (SQ with intervals of size \( 2r = 2 \))

\[
\delta = \frac{1}{2}
\]

Roger’s bound
- Theoretical upper bound for center density (last term being approximate)

\[
\log_2 \delta \leq \frac{N}{2} \log_2 \left( \frac{N}{4e\pi} \right) + \frac{1}{2} \log_2 \left( \frac{\pi N^3}{e^2} \right) + \frac{21}{4N + 10}
\]
The Vector Quantizer Advantage / Space-Filling Advantage

**Space-Filling Advantage: Densest Known Sphere Packings**

- Densest known packings for dimensions $N \leq 48$  [Conway, Sloane, 1998]
- Vertical axis: $\log_2 \delta + N(24 - N)/96$

![Diagram of densest known sphere packings](image)
### Space-Filling Advantage: Approximate SNR Gain

<table>
<thead>
<tr>
<th>dimension</th>
<th>densest packing</th>
<th>(name)</th>
<th>highest kissing number</th>
<th>approximate gain [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\mathbb{Z}$</td>
<td>Integer lattice</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$A_2$</td>
<td>Hexagonal lattice</td>
<td>6</td>
<td><strong>0.17</strong></td>
</tr>
<tr>
<td>3</td>
<td>$A_3 \simeq D_3$</td>
<td>Cuboidal lattice</td>
<td>12</td>
<td>0.29</td>
</tr>
<tr>
<td>4</td>
<td>$D_4$</td>
<td></td>
<td>24</td>
<td>0.39</td>
</tr>
<tr>
<td>5</td>
<td>$D_5$</td>
<td></td>
<td>40</td>
<td>0.47</td>
</tr>
<tr>
<td>6</td>
<td>$E_6$</td>
<td></td>
<td>72</td>
<td>0.54</td>
</tr>
<tr>
<td>7</td>
<td>$E_7$</td>
<td></td>
<td>126</td>
<td>0.60</td>
</tr>
<tr>
<td>8</td>
<td>$E_8$</td>
<td>Gosset lattice</td>
<td>240</td>
<td>0.66</td>
</tr>
<tr>
<td>9</td>
<td>$\Lambda_9$</td>
<td>Laminated lattice</td>
<td>240</td>
<td>0.70</td>
</tr>
<tr>
<td>10</td>
<td>$P_{10c}$</td>
<td>Non-lattice arrangement</td>
<td>336</td>
<td>0.74</td>
</tr>
<tr>
<td>12</td>
<td>$K_{12}$</td>
<td>Coxeter-Todd lattice</td>
<td>756</td>
<td>0.81</td>
</tr>
<tr>
<td>16</td>
<td>$BW_{16} \simeq \Lambda_{16}$</td>
<td>Barnes-Wall lattice</td>
<td>4320</td>
<td>0.91</td>
</tr>
<tr>
<td>24</td>
<td>$\Lambda_{24}$</td>
<td>Leech lattice</td>
<td>196560</td>
<td>1.04</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>1.35</td>
</tr>
<tr>
<td>$\infty$</td>
<td></td>
<td></td>
<td></td>
<td><strong>1.53</strong></td>
</tr>
</tbody>
</table>
Summary on Space-Filling Advantage

- Gain of unique to vector quantization: Packing of quantization cells in $N$-dimensional space
- Increases with quantizer dimension $N$
- Gain for $N \to \infty$: Difference between Shannon lower bound and ECSQ

![Graph showing space-filling gain (MSE) vs dimension $N$. The graph includes points at $N = 1, 2, 4, 8, 16, \infty$ with corresponding high-rate SNR gains.]
**Shape Advantage: Gaussian IID \((σ^2 = 1, \ R = 4)\)**

**Lloyd** \((N = 1)\)

\[
\frac{D_{N=2}}{D_{N=1}} \approx 0.81
\]

\[
ΔSNR \approx 0.93\, dB
\]

**LBG** \((N = 2)\)

**Shape Advantage of Vector Quantizers**

- Coding gain (0.93 dB for example) is larger than space-filling gain (0.17 dB for \(N = 2\))
- Vector quantizer can better adapt to shape of pdf (even without entropy coding)
Shape Advantage: Laplacian IID \( (\sigma^2 = 1, R = 4) \)

**Lloyd \((N = 1)\)**

\[
\frac{D_{N=2}}{D_{N=1}} \approx 0.64
\]

\[
\Delta \text{SNR} \approx 1.94 \text{dB}
\]

**LBG \((N = 2)\)**

**Shape Advantage of Vector Quantizers**

- Coding gain (1.94 dB for example) is larger than space-filling gain (0.17 dB for \(N = 2\))
- Vector quantizer can better adapt to shape of pdf (even without entropy coding)
Summary on Shape Advantage

- Gain of VQ due to exploitation of shape of pdf (without entropy coding)
- Overall gain for iid source: Space-filling gain + shape gain
- Shape advantage can also be exploited by entropy-constrained scalar quantization
The Vector Quantizer Advantage / Memory Advantage

Memory Advantage: Gauss-Markov \((\sigma^2 = 1, \varrho = 0.9, R = 2)\)

\[
\frac{D_{N=2}}{D_{N=1}} \approx 0.38
\]

\[
\Delta \text{SNR} \approx 4.20 \text{dB}
\]

Memory Advantage of Vector Quantizers

- Large coding gain (4.20 dB for example) for sources with memory
- Vector quantizer can exploit dependencies between samples
Memory Advantage: Gauss-Markov \((\sigma^2 = 1, \varrho = 0.9, R = 4)\)

Lloyd \((N = 1)\)

\[ \frac{D_{N=2}}{D_{N=1}} \approx 0.35 \]

\[ \Delta\text{SNR} \approx 4.55\,\text{dB} \]

Memory Advantage of Vector Quantizers

- Large coding gain (4.55 dB for example) for sources with memory
- Vector quantizer can exploit dependencies between samples
Summary on Memory Advantage

- Gain of VQ due to exploitation of dependencies between samples
- Largest gain to be made for sources with strong statistical dependencies
- Exploitation of memory advantage is one of the most relevant aspects in source coding
Optimal Vector Quantizer for Variable-Length Coding

Optimal Vector Quantizer with Consideration of Entropy Coding

- Similar to Scalar Entropy-Constrained Lloyd Quantizer
Optimal Vector Quantizer with Consideration of Entropy Coding

- Similar to Scalar Entropy-Constrained Lloyd Quantizer
- Minimization of Lagrangian cost for given Lagrange multiplier $\lambda$

$$J = D + \lambda \cdot R$$

$$= \frac{1}{N} \sum_{k} \int_{C_k} \|s - s'_k\|^2 f(s) \, ds + \frac{\lambda}{N} \sum_{k} \ell_k \int_{C_k} f(s) \, ds$$

Lagrange multiplier $\lambda > 0$ determines operation point (trade-off between rate and distortion)

Optimize Quantizer for given Lagrange multiplier $\lambda$

Derive necessary conditions for optimality (similar to EC Lloyd quantizer)

Construct iterative algorithm for designing quantizer

Similar as for EC Lloyd: Use large number of intervals in initialization
Optimal Vector Quantizer for Variable-Length Coding

Optimal Vector Quantizer with Consideration of Entropy Coding

- Similar to Scalar Entropy-Constrained Lloyd Quantizer
- Minimization of Lagrangian cost for given Lagrange multiplier $\lambda$

$$J = D + \lambda \cdot R$$

$$= \frac{1}{N} \sum_{\forall k} \int_{C_k} \| s - s'_k \|_2^2 f(s) \, ds + \frac{\lambda}{N} \sum_{\forall k} \ell_k \int_{C_k} f(s) \, ds$$

- Lagrange multiplier $\lambda > 0$ determines operation point (trade-off between rate and distortion)
Optimal Vector Quantizer with Consideration of Entropy Coding

Similar to Scalar Entropy-Constrained Lloyd Quantizer

Minimization of Lagrangian cost for given Lagrange multiplier $\lambda$

$$J = D + \lambda \cdot R$$

$$= \frac{1}{N} \sum_{k} \int_{C_k} \|s - s'_k\|_2^2 f(s) \, ds + \frac{\lambda}{N} \sum_{k} \ell_k \int_{C_k} f(s) \, ds$$

Lagrange multiplier $\lambda > 0$ determines operation point (trade-off between rate and distortion)

Optimize Quantizer for given Lagrange multiplier

Derive necessary conditions for optimality (similar to EC Lloyd quantizer)

Construct iterative algorithm for designing quantizer

Similar as for EC Lloyd: Use large number of intervals in initialization
Optimality Conditions for Variable-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $s_k'$, same as for LBG)

$$s_k' = \mathbb{E}\{ S \mid S \in C_k \} = \frac{1}{p_k} \int_{C_k} s f(s) \, ds$$

(training set: take average of assigned vectors)
Optimality Conditions for Variable-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $s'_k$, same as for LBG)

$$s'_k = \mathbb{E}\{ S \mid S \in C_k \} = \frac{1}{p_k} \int_{C_k} s f(s) \, ds$$

(training set: take average of assigned vectors)

2. Entropy condition (for codeword length $\ell_k$, same as for EC Llloyd)

$$\ell_k = - \log_2 p_k = - \log_2 \int_{C_k} f(s) \, ds$$

(training set: count assigned vectors)
Optimality Conditions for Variable-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $s'_k$, same as for LBG)

\[
s'_k = \mathbb{E}\{ S | S \in C_k \} = \frac{1}{p_k} \int_{C_k} s f(s) \, ds
\]

(training set: take average of assigned vectors)

2. Entropy condition (for codeword length $\ell_k$, same as for EC Llloyd)

\[
\ell_k = - \log_2 p_k = - \log_2 \int_{C_k} f(s) \, ds
\]

(training set: count assigned vectors)

3. Modified nearest neighbour condition (for quantization cells $C_k$ / encoder mapping $\alpha(.)$)

\[
\alpha(s) = \arg \min_{\forall k} \| s - s'_k \|_2^2 + \lambda \cdot \ell_k
\]
The Chou-Lookabaugh-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:
- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{s_n\}$ of considered source
The Chou-Lookabough-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:
- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{s_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$ and codeword length $\{\ell_k\}$
The Chou-Lookabough-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:
- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{s_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$ and codeword length $\{\ell_k\}$
2. Associate all vectors of the training set $\{s_n\}$ with one of the quantization cells $C_k$

$$q(s_n) = \arg \min_{\forall k} \|s_n - s'_k\|^2_2 + \lambda \cdot \ell_k$$

(modified nearest neighbor condition)
The Chou-Lookabough-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:
- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{s_n\}$ of considered source

**Iterative quantizer design**

1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$ and codeword length $\{\ell_k\}$
2. Associate all vectors of the training set $\{s_n\}$ with one of the quantization cells $C_k$

$$q(s_n) = \arg\min_{\forall k} \| s_n - s'_k \|_2^2 + \lambda \cdot \ell_k$$

(modified nearest neighbor condition)

3. Update the reconstruction vectors $\{s'_k\}$ and codeword length $\{\ell_k\}$ according to

$$s'_k = \frac{1}{n(k)} \sum_{\forall n: q(s_n) = k} s_n$$

and

$$\ell_k = -\log_2 \left( \frac{n(k)}{\sum_{\forall i} n(i)} \right)$$

where $n(k)$ is the number of sample vectors $s_n$ assigned to $C_k$
The Chou-Lookabough-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:
- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{s_n\}$ of considered source

Iterative quantizer design
1. Choose an initial set of $K$ reconstruction vectors $\{s'_k\}$ and codeword length $\{\ell_k\}$
2. Associate all vectors of the training set $\{s_n\}$ with one of the quantization cells $C_k$
   \[ q(s_n) = \arg \min_{\forall k} \|s_n - s'_k\|_2^2 + \lambda \cdot \ell_k \quad \text{(modified nearest neighbor condition)} \]
3. Update the reconstruction vectors $\{s'_k\}$ and codeword length $\{\ell_k\}$ according to
   \[ s'_k = \frac{1}{n(k)} \sum_{\forall n: q(s_n) = k} s_n \quad \text{and} \quad \ell_k = -\log_2 \left( \frac{n(k)}{\sum_{\forall i} n(i)} \right) \]
   where $n(k)$ is the number of sample vectors $s_n$ assigned to $C_k$
4. Repeat the previous two steps until convergence
Entropy-Constrained Vector Quantizer: Gaussian IID ($\sigma^2 = 1$, $R = 2$)

- **LBG ($N = 2$)**
  - $D = 0.107$
  - SNR = 9.69 dB

- **CLG ($N = 2$)**
  - $D = 0.086$
  - SNR = 10.68 dB

- **EC Lloyd ($N = 1$)**
  - $D = 0.089$
  - SNR = 10.51 dB

- Large gain (1.0 dB) relative to LBG algorithm (fixed-length coding)
- Gain relative to EC Lloyd reduces to space-filling gain (0.17 dB for $N = 2$)
Entropy-Constrained Vector Quantizer: Gaussian IID ($\sigma^2 = 1$, $R = 3$)

**LBG ($N = 2$)**

- $D = 0.0296$
- SNR = 15.29 dB

**CLG ($N = 2$)**

- $D = 0.0214$
- SNR = 16.70 dB

**EC Lloyd ($N = 1$)**

- $D = 0.0222$
- SNR = 16.53 dB

- Large gain (1.4 dB) relative to LBG algorithm (fixed-length coding)
- Gain relative to EC Lloyd reduces to space-filling gain (0.17 dB for $N = 2$)
Entropy-Constrained Vector Quantizer: Gauss-Markov ($\sigma^2 = 1, \varrho = 0.9, R = 3$)

**LBG ($N = 2$)**

- $D = 0.0125$
- SNR = 19.04 dB

**CLG ($N = 2$)**

- $D = 0.0099$
- SNR = 20.06 dB

**EC Lloyd ($N = 1$)**

- $D = 0.0222$
- SNR = 16.53 dB

- Large gain (1.0 dB) relative to LBG algorithm (fixed-length coding)
- Gain relative to EC Lloyd: Sum of memory gain and space-filling gain
Coding Efficiency of Vector Quantizers

**Vector Quantizer Advantages**

- Space-filling advantage
  
  Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \rightarrow \infty$)
Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - Unique to vector quantization: \(0.17\, \text{dB for } N = 2; \ 1.53\, \text{dB for } N \to \infty\)

- Shape advantage
  - Can also be exploited by entropy-constrained scalar quantization
Coding Efficiency of Vector Quantizers

**Vector Quantizer Advantages**

- **Space-filling advantage**
  - Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)

- **Shape advantage**
  - Can also be exploited by entropy-constrained scalar quantization

- **Memory advantage**
  - Can also be (partly) exploited by other coding techniques (topic of next lectures)
Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)
- Shape advantage
  - Can also be exploited by entropy-constrained scalar quantization
- Memory advantage
  - Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers
Coding Efficiency and Complexity / Summary of Observations

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - Unique to vector quantization: \(0.17\,\text{dB}\) for \(N = 2\); \(1.53\,\text{dB}\) for \(N \to \infty\)

- Shape advantage
  - Can also be exploited by entropy-constrained scalar quantization

- Memory advantage
  - Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers

- IID sources: Only space-filling gain (when comparing entropy-constrained designs)
Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \rightarrow \infty$)

- Shape advantage
  - Can also be exploited by entropy-constrained scalar quantization

- Memory advantage
  - Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers
- IID sources: Only space-filling gain (when comparing entropy-constrained designs)
- Sources with memory: Most important aspect is the memory advantage
Coding Efficiency of Vector Quantizers

**Vector Quantizer Advantages**

- Space-filling advantage
  - Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)
- Shape advantage
  - Can also be exploited by entropy-constrained scalar quantization
- Memory advantage
  - Can also be (partly) exploited by other coding techniques (topic of next lectures)

**Coding Efficiency of Vector Quantizers**

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers
- IID sources: Only space-filling gain (when comparing entropy-constrained designs)
- Sources with memory: Most important aspect is the memory advantage

⇒ Vector quantizers can asymptotically achieve rate-distortion bound for $N \to \infty$
Complexity of Vector Quantization

Decoding Complexity

- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large $N$
Complexity of Vector Quantization

Decoding Complexity
- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large $N$

Encoding Complexity
- Finding the “closest reconstruction vector” can become very complex
- Designing a good vector quantizer is already very complex
Complexity of Vector Quantization

Decoding Complexity
- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large $N$

Encoding Complexity
- Finding the “closest reconstruction vector” can become very complex
- Designing a good vector quantizer is already very complex

Usage of Vector Quantization
- Unconstrained vector quantizers are rarely used in practice
Complexity of Vector Quantization

Decoding Complexity
- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large $N$

Encoding Complexity
- Finding the “closest reconstruction vector” can become very complex
- Designing a good vector quantizer is already very complex

Usage of Vector Quantization
- Unconstrained vector quantizers are rarely used in practice
  ➔ Reduce complexity by imposing structural constraints
    - Tree-structured vector quantizers
    - Gain-shape vector quantizers
    - Lattice vector quantizers (important special case: Transform coding)
    - Trellis-coded quantization
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by \( N \) “basis vectors” \( \{b_k\} \)
  - Reconstruction vectors given by matrix of “basis vectors”

\[
s_{k_1, k_2, \ldots, k_N}' = M \cdot [k_1, k_2, \ldots, k_N]^T
\]

Simple decoder operation possible
Less complex encoding (can still be very complex for large \( N \))

Transform Coding

Lattice vector quantizer with orthonormal “basis vectors”
Very simple encoding and decoding

One of the most often used approaches in lossy coding
Will discuss in detail in next lectures
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by $N$ “basis vectors” $\{b_k\}$
  - Reconstruction vectors given by matrix of “basis vectors”
    \[
    s'_{k_1, k_2, \ldots, k_N} = M \cdot [k_1, k_2, \ldots, k_N]^T
    \]
- Simple decoder operation possible
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by $N$ “basis vectors” $\{b_k\}$
  - Reconstruction vectors given by matrix of “basis vectors”
    \[ s'_{k_1,k_2,\cdots,k_N} = M \cdot [k_1,k_2,\cdots,k_N]^T \]
- Simple decoder operation possible
- Less complex encoding (can still be very complex for large $N$)
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by $N$ “basis vectors” $\{b_k\}$
  - Reconstruction vectors given by matrix of “basis vectors”
    \[ s'_{k_1, k_2, \ldots, k_N} = M \cdot [k_1, k_2, \ldots, k_N]^T \]

- Simple decoder operation possible
- Less complex encoding (can still by very complex for large $N$)

Transform Coding

- Lattice vector quantizer with orthonormal “basis vectors”
- Very simple encoding and decoding
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by \( N \) “basis vectors” \( \{ b_k \} \)
  - Reconstruction vectors given by matrix of “basis vectors”
    \[
    s'_{k_1, k_2, \ldots, k_N} = M \cdot [k_1, k_2, \ldots, k_N]^T
    \]
- Simple decoder operation possible
- Less complex encoding (can still by very complex for large \( N \))

Transform Coding

- Lattice vector quantizer with orthonormal “basis vectors”
- Very simple encoding and decoding
  ➔ One of the most often used approaches in lossy coding
  ➔ Will discuss in detail in next lectures
Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process

- Two scalar quantizers

\[
\begin{align*}
Q_0 & \begin{cases}
0 & \rightarrow (0,1) \\
1 & \rightarrow (1,0)
\end{cases} \\
Q_1 & \begin{cases}
2 & \rightarrow (2,3) \\
3 & \rightarrow (3,2)
\end{cases}
\end{align*}
\]
Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process

- Two scalar quantizers + Procedure for switching between quantizers (state machine with $2^N$ states)

<table>
<thead>
<tr>
<th>state</th>
<th>quantizer</th>
<th>next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$Q_0$</td>
<td>$(A,B) \mapsto (0,1)$</td>
</tr>
<tr>
<td>1</td>
<td>$Q_1$</td>
<td>$(C,D) \mapsto (2,3)$</td>
</tr>
<tr>
<td>2</td>
<td>$Q_0$</td>
<td>$(A,B) \mapsto (1,0)$</td>
</tr>
<tr>
<td>3</td>
<td>$Q_1$</td>
<td>$(C,D) \mapsto (3,2)$</td>
</tr>
</tbody>
</table>
Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process

- Two scalar quantizers + Procedure for switching between quantizers (state machine with $2^N$ states)

<table>
<thead>
<tr>
<th>state</th>
<th>quantizer</th>
<th>next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$Q_0$</td>
<td>$(A,B) \mapsto (0,1)$</td>
</tr>
<tr>
<td>1</td>
<td>$Q_1$</td>
<td>$(C,D) \mapsto (2,3)$</td>
</tr>
<tr>
<td>2</td>
<td>$Q_0$</td>
<td>$(A,B) \mapsto (1,0)$</td>
</tr>
<tr>
<td>3</td>
<td>$Q_1$</td>
<td>$(C,D) \mapsto (3,2)$</td>
</tr>
</tbody>
</table>

Encoding Process

- Trellis formulation of possible quantizer switching

state 0 →

state 1 →

state 2 →

state 3 →
Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process
- Two scalar quantizers + Procedure for switching between quantizers (state machine with $2^N$ states)

Encoding Process
- Trellis formulation of possible quantizer switching  ➔ Viterbi algorithm
Example: TCQ Performance for Gaussian IID

- Gish & Pierce (1.53 dB)
- Entropy-constrained TCQ (4 states) ≈ 1 dB
- Entropy-constrained TCQ (8 states)
- Entropy-constrained TCQ (64 states)

SNR loss relative to rate-distortion function [dB]

Entropy-constrained scalar quantizer

Bit rate [bps]
Example: TCQ Performance for Gaussian IID

SNR loss relative to rate-distortion function [dB]

Gish & Pierce (1.53 dB)

entropy-constrained scalar quantizer

bit rate [bps]

0 0.5 1 1.5 2

0 0.5 1 1.5

0 0.5 1 1.5 2
Example: TCQ Performance for Gaussian IID

![Graph showing the SNR loss relative to rate-distortion function in dB against bit rate in bps. The graph compares different quantization methods, including entropy-constrained scalar quantizer and entropy-constrained TCQ with 4 states and 64 states. The graph includes a note indicating the performance of Gish & Pierce (1.53 dB).]
Example: TCQ Performance for Gaussian IID

SNR loss relative to rate-distortion function [dB]

Gish & Pierce (1.53 dB)

entropy-constrained scalar quantizer

entropy-constrained TCQ (4 states)

≈ 1 dB
Example: TCQ Performance for Gaussian IID

![Graph showing SNR loss relative to rate-distortion function vs. bit rate for different TCQ configurations.

- Gish & Pierce (1.53 dB)
- Entropy-constrained scalar quantizer
- Entropy-constrained TCQ (4 states)
- Entropy-constrained TCQ (8 states)

Approximately 1 dB difference between the entropy-constrained scalar quantizer and the Gish & Pierce configuration.]
Example: TCQ Performance for Gaussian IID

- Gish & Pierce (1.53 dB)
- Entropy-constrained scalar quantizer
- Entropy-constrained TCQ (4 states)
- Entropy-constrained TCQ (8 states)
- Entropy-constrained TCQ (64 states)

SNR loss relative to rate-distortion function [dB]

≈ 1 dB
Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions $N$
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers ($1.53$ dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?

No! – Complexity of vector quantization is a serious issue!

Require lossy coding techniques with high rate-distortion efficiency and a complexity suitable for wide range of implementations

Particularly important:

- Exploitation of dependencies between samples!
Summary of Lecture

Vector Quantization (VQ)
- Straightforward extension of scalar quantization to higher dimensions $N$
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages
- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques
Summary

Summary of Lecture

Vector Quantization (VQ)
- Straightforward extension of scalar quantization to higher dimensions $N$
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages
- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \rightarrow \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?
Summary of Lecture

Vector Quantization (VQ)
- Straightforward extension of scalar quantization to higher dimensions $N$
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages
- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?
- No! – Complexity of vector quantization is a serious issue!
Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions $N$
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?

➡ No! – Complexity of vector quantization is a serious issue!
→ Require lossy coding techniques with high rate-distortion efficiency and a complexity suitable for wide range of implementations
Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions $N$
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers ($1.53$ dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?

- No! – Complexity of vector quantization is a serious issue!
- Require lossy coding techniques with high rate-distortion efficiency and a complexity suitable for wide range of implementations
- Particularly important: Exploitation of dependencies between samples!
Exercise 1: Space-Filling Gain for 2-dimensional Vector Quantizer

Calculate the gain (in signal-to-noise ratio) of optimal 2-dimensional vector quantization relative to optimal scalar quantization for high rates on the example of a uniform pdf.

Hints:

- In two dimensions, the optimal quantization cells are regular hexagons; the associated reconstruction vectors are located in the centers of the hexagons.
- For high rates, border effects can be neglected. It can be assumed that the signal space for which the pdf is non-zero is completely filled with regular quantization cells.